ATIS-1000667.2002(R2012)

Intelligent Network

AMERICAN NATIONAL STANDARD FOR TELECOMMUNICATIONS

ATIS is the leading technical planning and standards development organization committed to the rapid development of global, market-driven standards for the information, entertainment and communications industry. More than 200 companies actively formulate standards in ATIS' Committees and Forums, covering issues including: IPTV, Cloud Services, Energy Efficiency, IP-Based and Wireless Technologies, Quality of Service, Billing and Operational Support, Emergency Services, Architectural Platforms and Emerging Networks. In addition, numerous Incubators, Focus and Exploratory Groups address evolving industry priorities including Smart Grid, Machine-to-Machine, Connected Vehicle, IP Downloadable Security, Policy Management and Network Optimization.

ATIS is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), 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). ATIS is accredited by the American National Standards Institute (ANSI). For more information, please 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 OFMERCHANTABILITY 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.

ATIS-1000667.2002(R2012), *Intelligent Network*

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

Published by

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

Printed in the United States of America.

ATIS-1000667.2002 (R2012)

(Formerly T1.667-2002, Revision of T1.667-1999)

American National Standard for Telecommunications

Intelligent Network

Secretariat

Alliance for Telecommunications Industry Solutions

Approved May 23, 2002

American National Standards Institute, Inc.

Abstract

This standard establishes an architectural framework in which the model of the Intelligent Network (IN) is defined. The architecture is intended to provide the flexibility to support a wide range of services and facilitates the evolution of future IN functional capabilities through its evolvable, modular structure to achieve service independence.

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, 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.

Work on the standardization of the components and capabilities of the Intelligent Network (IN) have been underway since 1989. The U.S. has been instrumental in advancing international IN standards. This standard is based on material adapted from ITU-T Recommendations in the Q.12xx series, and incorporates provisions to meet the unique needs of the North American telecommunications environment and to support service standardization initiatives of Committee T1 requiring standardized IN capabilities and signaling.

This standard is based on the merger of T1.667-1999 and T1.667a-2000 (Amendments). In addition, some of the more significant changes are:

- Support of multiple routing numbers;
- Support of Timeout and O_DTMF_Entered events;
- Support of emergency calling service;
- Support of direct SCF-SRF communications;
- Removal of Annex B.

This standard was prepared by Working Group T1S1.7 on Services, Architecture, and Control of T1S1, the Technical Subcommittee on Services, Architectures and Signaling.

There are five annexes to this standard. Annex A is normative and is considered part of this standard; Annexes B through E are informative and are not considered part of this standard. Similarly, footnotes are not part of this standard.

Future control of this document will reside with Accredited Standards Committee — Telecommunications, T1. This control of additions to the specification, such as protocol evolution, new applications, and operational requirements, will permit compatibility among U.S. networks. Such additions will be incorporated in an orderly manner with due consideration to the ITU-T layered model principles, conventions, and functional boundaries.

Suggestions for improvement of this standard will be welcome. These should be sent to the Alliance for Telecommunications Industry Solutions, T1 Secretariat, 1200 G Street, NW, Suite 500, Washington DC 20005.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Telecommunications, T1. Committee approval of the standard does not necessarily imply that all committee members voted for its approval. At the time, it approved this standard; the T1 Committee had the following members:

Organization Represented	Name of Representative
Cable & Wireless	Olga Aparicio Roderick Dottin (Alt.)
Covad Communications Co.	Ron Marquardt David Rosenstein (Alt.)
Qwest	James L. Eitel Richard Prince (Alt.)
Rhythms	Rand Kennedy David Reilly (Alt.)
Rogers Wireless Inc.	Edward O'Leary Peter Oldfield (Alt.)
SBC Communications, Inc.	Chuck Bailey Bob Hall (Alt.)
Sprint - Local Telecom. Division	Leroy D. Kellogg
US Telecom Association (USTA)	Paul Hart Donald G. Bender (Alt.)
Verizon Communications	Josephine Gallagher James F. Baskin (Alt.)

GENERAL INTEREST

Organization Represented	Name of Representative
BOPS Inc.	Ali S. Sadri, PhD
CSI Telecommunications	Michael S. Newman Thomas G. Croda (Alt.)
Cingular Wireless LLC	Don Zelmer Mark Grant (Alt.)
Defense Information Systems Agency	Don Choi
Golden Bridge Technology Inc.	Kourosh Parsa Karin Zickermann (Alt.)
LightSurf Technologies, Inc.	John Hansen Shekhar Kirani (Alt.)
Microcell Connexions	Venkatesh Sampath Besma Smida (Alt.)
National Communications System	Nicholas Andre F. McClelland (Alt.)
NTIA	Neal B. Seitz
Quintessent Communications Inc.	Dave Deutschman
Rural Utilities Service	Orren E. Cameron III Norberto Esteves (Alt.)
Tantivy Communications	Mel Woinsky Clifton J. Barber (Alt.)
Telcordia Technologies	Rick Harrison Cliff Halevi (Alt.)
Voicestream Wireless Corp.	Gary K. Jones Mark Younge (Alt.)

INTEREXCHANGE CARRIERS

Organization Represented	Name of Representative
AT&T	Charles A. Dvorak Percy Tarapore (Alt.)
Bell Canada	P. Norman Smith
Intelsat	Mark T. Neibert
Lockheed Martin Global Telecom	Prakash Chitre
Sprint - Long Distance Division	James Lord Al White (Alt.)
WorldCom	Yi-Shang Shen J. Martin Carroll (Alt.)

MANUFACTURERS

Organization Represented	Name of Representative
3COM	Fred Lucas James Renkel (Alt.)
Acterna	Michael Lewis Dick Bobilin (Alt.)
ADC Telecommunications Inc.	Nelson Zagalsky
Alcatel USA Inc.	Ken Biholar Cheri Dickerson (Alt.)
Aware, Inc.	Marcos Tzannes Richard L. Stuart (Alt.)
Beatnik, Inc.	Chris Grigg Chris Muir (Alt.)
Broadcom Corp.	Vladimir Oksman
Catena Networks Inc.	Ed Eckert Andy Weirich (Alt.)
Centillium Communications, Inc.	Dr. Syed Abbas Guozhu Long (Alt.)
Cisco Systems, Inc.	John McDonough John Krahner (Alt.)
Conexant Systems, Inc.	Quentin C. Cassen Arun Arunachalam (Alt.)
Copper Mountain Networks	John Reister Jack Yang (Alt.)
dynamicsoft Inc.	Dean Willis Jonathan Rosenberg (Alt.)
Elastic Networks, Inc.	Patrick H. Stanley, P.E.
Ericsson Inc.	Bob Slocum Asok Chatterjee (Alt.)
Excelsus Technologies Inc.	William J. Buckley Don Robert House (Alt.)
Fujitsu America Inc.	Mike White Hirohiko Yamamoto (Alt.)
Globespan Semiconductor, Inc.	Massimo Sorbara Clete Gardenhour (Alt.)
Harris Corp.	Marlis Humphrey
Hekimian Laboratories	William H. Duncan
Hewlett-Packard	Steve Mills Karen Higginbottom (Alt.)
Hughes Network Systems, Inc.	Dr. Leonard Golding Enrique Laborde (Alt.)

Organization Represented	Name of Representative
IBM Corp.	Jeff H. Derby Evangelos Eleftheriou (Alt.)
Juniper Networks, Inc.	Elizabeth Lytle
Lucent Technologies	Greg Ratta Rick Townsend (Alt.)
Luxxon Corp.	Tao Lin Bert Rackett (Alt.)
Marconi Communications	Mark Scott
Metawave Communications Corp.	Shimon Scherzer
Motorola Inc.	Ken Skurnak Syed Niaz (Alt.)
NEC America Inc.	Dia Helmy
Next Level Communications	Sabit Say Todd Pett (Alt.)
Nokia Telecommunications Inc.	Chris Wallace Margaret Livingston (Alt.)
Nortel Networks	Subhash Patel Joseph A. Zebarth (Alt.)
Octasic, Inc.	Serge Fournier Thomas Awad (Alt.)
Ocular Networks, Inc.	Ron Fang Chris Roller (Alt.)
OKI America Inc.	Henri Suyderhoud Hisao Fujikawa (Alt.)
Paradyne Corp.	Phil Kyees
PMC-Sierra, Inc.	Steve Gorshe
Qualcomm Inc.	Mark Epstein Ed Tiedemann (Alt.)
Siemens Information & Communications Networks, Inc.	David E. Francisco Jim Stanco (Alt.)
SnowShore Networks, Inc.	Eric Burger Doug Corrigan (Alt.)
ST Microelectronics	Raffaele Penazzi Stefania Boiocchi (Alt.)
Symmetricom Inc.	Don Skipwith Ed Butterline (Alt.)
Tellabs Operations, Inc.	Tom Rarick William A. Walker (Alt.)
Tellium, Inc.	Krishna Bala, PhD Siegfried Giebl (Alt.)
TELOS Technology	Steve Williams Brad Long (Alt.)
Texas Instruments	Thomas Maudoux Krista Jacobsen (Alt.)
TranSwitch Corp.	Jitender Vij Edwin Soltysiak (Alt.)
TruePosition Inc.	Matthew Ward Thomas Ginter (Alt.)
Voyan Technology	Michail Tsatsanis Rolf Fiebrich (Alt.)
Westell Technologies, Inc.	Guy Cerulli Gordon Purtell (Alt.)
Zarlink Semiconductor	Maamoun AbouSeido Gary Jin (Alt.)

Subcommittee T1S1 on Services, Architectures, and Signaling, which developed this standard, had the following members:

B. Hall, T1S1 Chair

G. Ratta, T1S1 Vice-Chair

Organization Represented	Name of Representative
Acterna	Michael Lewis Dick Bobilin (Alt.)
Alcatel USA Inc.	Jeff Copley
AT&T	Martin Dolly Janey Cheu (Alt.)
AT&T Wireless Services, Inc.	Peter Musgrove Brian Daly (Alt.)
Bell Canada	Stewart Patch P. Norman Smith (Alt.)
BellSouth Telecommunications Inc.	Robert V. Epley David Whitney (Alt.)
C.S.I. Telecommunications	Michael S. Newman Thomas G. Croda (Alt.)
Cisco Systems	Rajiv Kapoor Chip Sharp (Alt.)
Compaq Computer Corp.	John L. Schantz Steve Upton (Alt.)
Defense Info. Systems Agency	Don Choi Ralph Liguori (Alt.)
Ericsson Inc.	Bob Slocum
Fujitsu America Inc.	Mark Stewart Doug Hunt (Alt.)
Harris Corp.	Marlis Humphrey
Hekimian Laboratories	William H. Duncan
Inet Technologies Inc.	Jarrett Archer Trevor Schelp (Alt.)

Organization Represented	Name of Representative
Intelsat	Mark T. Neibert
Lucent Technologies	Greg Ratta
National Communications System	Nicholas Andre H. Folts (Alt.)
Nortel Networks	Subhash Patel Joseph A. Zebarth (Alt.)
OKI America Inc.	Henri Suyderhoud Hisao Fujikawa (Alt.)
Oresis Communications, Inc.	Michael R. Zeug George Shenoda (Alt.)
Paradyne Corp.	Phil Kyees
Qwest	Steve Showell James L. Eitel (Alt.)
Rhythms	Rand Kennedy David Reilly (Alt.)

Organization Represented	Name of Representative
SBC Communications, Inc.	B.S. Sambasivan Bob Hall (Alt.)
Siemens Information and Communication Networks, Inc.	Rajendra Udeshi Ron Franks (Alt.)
Sprint - Long Distance Division	James Lord
Telcordia Technologies	Wesley Downum Selvan Rengasami (Alt.)
Tellabs Operations, Inc.	Brian Yarger Mike Wurst (Alt.)
US Telephone Association - USTA	Paul Johnson Donald G. Bender (Alt.)
Verizon Communications	Dana Shillingburg
Worldcom	Yatendra Pathak Bernard Ku (Alt.)

Working Group T1S1.7 on Services, Architecture, and Control developed this standard. Over the course of its development, the following individuals participated in the Working Group's discussions and made significant contributions to the standard:

Martin Dolly, Working Group Chair

Joe Zebarth, Working Group Vice-Chair

Tung-Hai Hsiao, Editor

Cheryl Toth, Editor

T1S1.7 Working Group members:

Khalid Ahmad	Zahia Aidoud	George Allen	Nick Andre	Bjorn Ahle
Dawn Benedetto	Ken Biholar	John Boal	Dick Bobilin	Rolfe Buhrke
Jim Calme	Joanie Cardone	Janey Cheu	Don Choi	Koan Chong
Michael Collison	Jeff Copley	Yogesh Dave	Ranga Dendi	Martin Dolly
Wesley Downum	Julie Federico	Hui Feng	Debi Ferguson	Fran Fischbach
Hal Folts	Larry Forni	Brian Foster	Gilberto Frederick	Fred Gaechter
Andrew Gallant	Jim Garrahan	Stuart Goldman	Eric Gray	Dan Greene
Kevin Hager	Robert Hall	Rich Hemmeter	Jay Hilton	Mark Hosford
Doug Hunt	Syed Zaeem Hosain	Tung-Hai Hsiao	Donna Imrich	Rajiv Kapoor
Ghassem Koleyni	Tmima Koren	Bill Kwan	Doris Lebovits	D Leung
William Lipp	Sandra Lopez	Jim Lord	Norb Lucash	Kieth Mainwaring
Patti McCalmont	Karen McCourt	Michael McGrew	Mike McLoughlin	Vladimir Mikhailov
Farshid Mohammadi	A.J. Moore	Enrique Moviles	Ananth Nagarajan	Mark Neibert
Bruce Northcote	Mart Nurmet	Hans Oudmaijer	Stewart Patch	Yatendra Pathak
Greg Ratta	Selvam Rengasami	Lewis Robart	Larry Ruegger	Sam Sambasivan
Niranjan Sandesara	John Schantz	Viquar Shaikh	Chip Sharp	Dana Shillingburg
Steve Showell	Ray Singh	Bob Slocum	Carl Smedberg	Mark Stewart
Art Stoutenburg	Andy Sudduth	Henri Suyderhoud	Carolyn Taylor	Kamlesh Tewani

Bruce Thompson	Mike Tisiker	Cheryl Toth	Rick Townsend	Jean Trakinat
Linda Troy	Rajendra Udeshi	Dhadesugoor Vaman	Dave VanderMeiden	Al Varney
Stan Wainberg	Robert Waller	Tom Walsh	Dick Weadon	Albert White
David Whiteford	Ping Wong	Brian Yarger	Wing-Cheong Yeung	Albert Yuhan
Sy Yules	Joe Zebarth	Wayne Zeuch	Michael Zeug	

Table of Contents

Table of Tables

Table of Figures

American National Standard for Telecommunications –

Intelligent Network

1 Scope, purpose and application

This American National Standard defines Intelligent Network (IN) capabilities for telecommunications networks. This standard shall hereinafter be referred to as the T1 IN Standard or T1 IN for shorthand. This document establishes an architectural framework in which the model of the Intelligent Network is defined. The architecture is intended to provide the flexibility to support a wide range of services and facilitates the evolution of future IN functional capabilities through its evolvable, modular structure to achieve service independence. The structure is also intended to support the multi-vendor environment and internetwork capabilities needed to make IN services globally available.

The information contained in this standard is based upon capabilities defined in the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) Intelligent Network Capability Set 1 (IN CS-1 1995) Recommendations, as well as on the additional protocol needed to support the T1 IN Call Model. The scope of this document includes refinements and selection of options to the existing ITU-T Recommendations to support North American implementations of IN. The document also takes into account the unique needs of the North American telecommunication environment. . Refer to Annex E for a description of supplemental references.

IN CS-1 defines the IN architecture using a four-plane model consisting of:

- Service Plane
- Global Functional Plane
- Distributed Functional Plane
- Physical Plane

While this model is essential for fully describing service development through service execution, this standard focuses on the distributed functional plane and the physical plane. The primary rationale for this is that the implementation of the upper two planes has been vendor-specific in the North American marketplace. Management of the services and data within T1 IN is not part of the scope of this standardization effort.

From an access perspective, T1 IN is limited to narrowband connections (e.g., POTS, BRI). T1 IN is intended to be able to inter-work with all North American narrowband trunk and line signaling. Internetworking between multiple PSTNs, and between PSTNs and wireless will be supported. Initially, as in IN CS-1, calls from the switching elements are limited to single ended, single point of control.

This Standard defines the INAP (intelligent network application protocol) required for support of T1 IN. It supports interactions between the following four functional entities (FEs) as defined in clause 5 of this T1 IN Standard:

- Service Switching Function (SSF)
- Service Control Function (SCF)
- Specialized Resource Function (SRF)

- Service Data Function (SDF)

The scope of this T1 IN Standard is the further development of the INAP for both the Integrated Services Digital Network (ISDN) and the Public Switched Telecommunications Network (PSTN).

The T1 IN Standard is intended as a guide to implementers and network operators to ensure interworking between different vendors' equipment for all the T1 IN defined interfaces (SCF-SSF, SCF-SRF and SCF-SDF).

The IN Application Protocol defines those functions needed by the T1 IN Standard.

1.1 Motivation

The term, Intelligent Network (IN), is used to describe an architectural concept that is intended to be applicable to all telecommunications networks. IN aims to facilitate the introduction of new services (e.g., Universal Personal Telecommunication (UPT), Virtual Private Network (VPN), Freephone, etc.) based on greater flexibility and new capabilities.

IN Recommendations are motivated by the interests of telecommunication services providers to rapidly, cost effectively and differentially satisfy their existing and potential market needs for services. Also, these service providers seek to improve the quality and reduce the cost of network service operations and management.

Additionally, current trends in technology permit a greater degree of intelligence and greater freedom in the allocation of intelligence in the telecommunications network. For example, the improved mobility derived from miniaturization of electronic components allows for a greater degree of distributed functionality within and between service provider networks. Factors permitting such intelligence include: advances in digital transmission and switching, common channel signaling, distributed data processing, data base management and expert systems.

1.2 Objectives of Intelligent Network

The objective of IN is to allow the inclusion of additional capabilities to facilitate provisioning of service, independent of the service/network implementation in a multi-vendor environment. Service implementation independence allows service providers to define their own services independent of service-specific developments by equipment vendors.

Network implementation independence allows network operators to allocate functionality and resources within their networks and to efficiently manage their networks independent of network implementation-specific developments by equipment vendors.

1.3 Scope of Intelligent Network

Types of networks: IN is applicable to a wide variety of networks, including but not limited to: public switched telephone network (PSTN), mobile, packet switched public data network (PSPDN) and integrated services digital network (ISDN) – both narrow-band-ISDN (N-ISDN) and broadband-ISDN (B ISDN).

Type of services: IN supports a wide variety of services, including supplementary services, and utilizes existing and future bearer services (e.g., as those defined in N-ISDN and B-ISDN contexts).

The scope of the IN distributed functional plane (DFP) architecture for T1 IN is driven by service requirements, and constrained by the capabilities of the embedded base of evolvable network technology. The scope of functionality required to support desired services includes functionality to provide:

- end user access to call/service processing;
- service invocation and control;
- end user interaction with service control;
- and service management.

The scope of each of these aspects is addressed below.

1.3.1 End user access

End user access to call/service processing for T1 IN will be provided via the following access arrangements:

- analog line interfaces;
- ISDN BRI and PRI; and
- traditional trunk and SS7 interfaces.

Note that this does not preclude the use of these interfaces to support access from private or mobile networks.

1.3.2 Service invocation and control

Call/service processing for T1 IN builds upon the current call-processing infrastructure of existing digital exchanges. It does so by using a generic model of existing call control functionality to process basic two-party calls, then adding service switching functionality to invoke and manage IN service logic. Once invoked, IN service logic is executed under the control of service control functionality, in conjunction with service data functionality. With this distributed approach to call/service processing, the existing call control functionality retains ultimate responsibility for the integrity of calls, as well as for the control of call processing resources. The following call/service processing constraints apply:

A. Call control and service switching functionality are tightly coupled; thus, the relationship between SSF and CCF is not standardized.

B. A call is either between two or more end users that are external to the network and addressable via a directory number or combination of directory number and bearer capability, or a call is between one or more end users and the network itself.

C. A call may be initiated by an end user, or by an SCF within the network on behalf of an end user. To supplement a call, IN service logic may either be invoked by an end user served by an IN exchange, or by the network on behalf of an end user.

D. A call may span multiple exchanges. As such, each exchange only controls the portion of the call in that exchange – call processing is functionally separated between exchanges. IN service logic invoked on IN exchanges in such an inter-exchange call are managed independently by each IN exchange.

E. Existing exchanges can be viewed as having two functionally separate sets of call processing logic that coordinate call processing activities to create and maintain a basic two-party call. This functional separation is provided between the originating portion of the call and the terminating portion of the call. This functional separation should be maintained in an IN exchange to allow IN service logic invoked on the originating portion of the call (i.e., on behalf of the calling party) to be managed independently of IN service logic invoked on the terminating portion of the call (i.e., on behalf of the called party).

F. It is desirable to allow multiple IN-supported service logic instances to be simultaneously active for a given end user. It is also recognized that non-IN service logic will continue to exist in the network. As such, service feature logic instances mechanisms for T1 IN should:

- determine which service logic to invoke for a given service request. This mechanism should select the appropriate IN-supported service logic or non-IN-supported service logic, and block the invocation of any other service logic for that particular service request;
- limit simultaneously active IN- and non-IN-supported service logic instances;
- ensure that simultaneously active IN-supported service logic instances adhere to the single-ended, single point of control restriction.

G. The distributed approach and added complexity of call/service processing for T1 IN requires mechanisms for fault detection and recovery, allowing graceful termination of calls and appropriate treatments for end users.

1.3.3 End user interaction

End user interaction with the network to send and receive information is provided by service switching and call control resources, augmented by specialized resources. These specialized resources are controlled by service control functionality, and are connected to end users via call control and service switching functionality.

1.3.4 Service management

Service management functionality is used to provision and manage the service control functionality, service data functionality, and specialized resource functionality in the network, outside the context of call/service processing. Standardized interfaces for this functionality are outside the scope of T1 IN. However, the ability of a service subscriber to interact directly with subscriber-specific service management information is not excluded or constrained for T1 IN.

1.4 Scope of Intelligent Network physical plane

The scope of the IN Physical plane of the IN architecture for T1 IN is consistent with the IN conceptual model. The physical plane identifies the different physical entities and the interfaces between these entities. It shall also meet the following criteria:

- service implementation independence;
- network implementation independence;
- vendor and technology independence.

2 Normative references

The following standards contain provisions, which through reference in this text, constitute provisions of this American National Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision and parties to agreements based on this American National Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

T1.111-2001, *Signaling System No.7, Message Transfer Part* [1]

T1.112-1996, *Telecommunications - Signalling System No. 7 - Signaling Connection Control Part (SCCP)*[1]

T1.113-2000, *Telecommunications - Signalling System No. 7 - Integrated Services Digital Network (ISDN) User Part*[1]

T1.114-2000, *Telecommunications - Signalling System No. 7 (SS7) - Transaction Capability Application Part (TCAP)*[1]

T1.607-2000, *Telecommunications - Integrated Services Digital Network (ISDN) - Layer 3 Signaling Specification for Circuit-Switched Bearer Service for Digital Subscriber Signaling System No. 1 (DSS1)* [1]

T1.610-1998, *Telecommunications - Generic Procedures for the Control of ISDN Supplementary Services*[1]

T1.628-2000, *Emergency Calling Service* [1]

T1.701-1994 (R1999), *Telecommunications - Universal Personal Telecommunication (UPT) - Service Description (Service Set One)*[1]

T1.702-1995 (R1999), *Telecommunications - Personal Communications Technology*[1]

ITU-T Recommendation E.164 (05/97), *The international public telecommunication numbering plan* [2]

ITU-T Recommendation Q.763 (12/99), *Signalling System No. 7 - ISDN User Part formats and codes*[2]

ITU-T Recommendation Q.931 (05/98), *ISDN user-network interface layer 3 specification for basic call control*[2]

ITU-T Recommendation Q.773 (06/97), *Signalling System No. 7 - Transaction capabilities formats and encoding)*[2]

ITU-T Recommendation X.200 (07/94), *Information technology - Open Systems Interconnection – Basic reference model 2)*[2]

ITU-T Recommendation X.680 (07/02) | ISO/IEC 8824-1:1994, *Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation 2)*[2]

---

[1] This document is available from the Alliance for Telecommunications Industry Solutions, 1200 G Street N.W., Suite 500, Washington, DC 20005. <http://www.atis.org>
[2] This document is available from the International Telecommunications Union. < http://www.itu.int/ITU-T/ >

ITU-T Recommendation X.681 (07/02) | ISO/IEC 8824-2:1994, *Information technology - Abstract Syntax Notation One (ASN.1): Information object specification 2)*[2]

ITU-T Recommendation X.682 (07/02) | ISO/IEC 8824-3:1994, *Information technology - Abstract Syntax Notation One (ASN.1): Constraint specification 2)*[2]

ITU-T Recommendation X.683 (07/03) | ISO/IEC 8824-4:1994, *Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications 2)*[2]

ITU-T Recommendation X.690 (07/02) | ISO/IEC 8825-1:1994, *Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER) 2)*[2]

ITU-T Recommendation X.880 (07/94) | *ISO/IEC 13712-1:1994, Information technology - Remote Operations: Concepts, model and notation 2)*[2]

(For extensive informative references, see Bibliography)

3 Definitions

This clause provides a glossary of terms and definitions studied for application in the documentation of Intelligent Networks.

To the extent practicable, terms that have been defined previously are used unchanged and reference to the source of the definition is shown next to the term in parentheses. The definitions that have been changed to make them appropriate for this application are considered to be new definitions; however, reference to the source definition is also shown in parentheses.

3.1 Terms and definitions

For the purpose of this standard, the following definitions apply:

3.1.1 access channel: A designated part of the information transfer capability having specified characteristics, provided at the user-network interface.

3.1.2 access code: A code(s) for “customized numbering plan(s)”: attendant access, escapes to public network, etc.

3.1.3 access function: A set of processes in a network that provide for interaction between the user and a network.

3.1.4 adjunct (AD): An entity in the intelligent network that is functionally equivalent to a service control point but is directly connected to a service switching point.

3.1.5 agent: An entity acting on behalf of another.

3.1.6 alerting pattern: a specific pattern used to alert a subscriber (e.g., distinctive ringing, tones, etc.).

3.1.7 application context (AC): An Application Context describes the functions which are to be used for a particular instance of communication,

3.1.8 application context negotiation (ACN): Context negotiation establishes at the beginning of a transaction that set of user protocol ASEs will be potentially exchanged during a transaction.

3.1.9 application entity (AE): The system-independent application activities that are made available as application services to the application agent, e.g., a set of application service elements that together perform all or part of the communication aspects of an application process.

3.1.10 application entity invocation (AEI): The actual “run-time” program that performs all or a subset of the communication functions defined by the AE-type specifications.

3.1.11 application program: Logic residing in the service control and service management realms that directs and/or controls the performance of actions in the network to provide and/or manage the provision of IN service features.

3.1.12 application programming interfaces (APIs): Interfaces that support the process of creating, installing, testing, modifying, etc., IN application programs.

3.1.13 application protocol data unit (APDU): A unit of application data specified in an application protocol and consisting of application protocol control information and application protocol user data.

3.1.14 application service element (ASE): A coherent set of integrated functions to help accomplish application communication, e.g., within an application entity.

3.1.15 attribute: Information of a particular type. Refer to ITU-T Recommendation X.501.

3.1.16 basic call: A call between two users that does not include additional features (e.g., a plain telephone call).

3.1.17 basic call process (BCP): The sequence of activities used in processing a basic call attempt.

3.1.18 basic call state model (BCSM): A high-level finite state machine model of call processing for basic call control (i.e., a two-party non-IN call). The model might only cover a portion of a call attempt, e.g., an originating BCSM or terminating BCSM, or the whole attempted call connection, originating user to terminating user.

3.1.19 basic rate interface (BRI): A user-network access arrangement that corresponds to the interface structure composed of two B-channels and one D-channel. The D-channel for this type of access is 16 kbit/s, while the B-Channel rate is 64 kbit/s.

3.1.20 bearer control: The set of functions used to direct the low layer (common) means of transmission.

3.1.21 bind: A mechanism used during Association Control for authentication. Refer to ITU-T Recommendation X.500.

3.1.22 business group identity (BGID): The Basic Business Group or Multiswitch Business Group Identity of the calling/called party, e.g., group based services.

3.1.23 call: The use, or possible use, of one or more connections set up between two or more users and/or service(s).

3.1.24 call control: The set of functions used to process a call (e.g., provide service features and establish, supervise, maintain and release connections).

3.1.25 call control agent functional entity (CCAF): A functional entity that provides network access functions for users, interacting with call control functional entities in providing services.

3.1.26 call control function (CCF): The network intelligence that provides call/connection processing and control.

3.1.27 call control functional entity: Functional entities that cooperate with each other to provide network call processing functions.

3.1.28 call manager (CM): The entity in the SSF that provides mechanisms to support multiple concurrent instances of IN service logic instances and non-IN service logic instances on a single call.

3.1.29 call model: A representation of functions involved in processing a call.

3.1.30 call/service processing: The execution of logic by a switching or control function to advance a call attempt or a service request.

3.1.31 call segment: A specific portion of the processing of a call.

3.1.32 called party: The entity that is the recipient of a call.

3.1.33 calling party: The entity that originates the call.

3.1.34 capability set (CS): A set of intelligent network capabilities that are to be the subjects of ITU-T standardization activities and for which the availability of ITU-T Recommendations will be targeted for a particular time frame.

3.1.35 carrier (commercial telecommunications): The organization whose function is to provide the particular service, e.g., an agent of a private network/facility, intra-serving area, or a specific inter-exchange carrier or international carrier.

3.1.36 carrier access code (CAC): A code used to select a carrier.

3.1.37 carrier identification code: Identifies the carrier.

3.1.38 carrier selection: Identifies whether the caller dialed the selected carrier and whether the caller presubscribed to the selected carrier.

3.1.39 charge number: The number to be charged for the call.

3.1.40 class of service: This is either a Customer Class of Service, a Trunk Class of Service, or a Private-Facility Class of Service. It may refer to either originating or terminating accesses.

3.1.41 collected address information: An interpretation of the dialed digits specifying the nature of address and the *Called Party Number*.

3.1.42 collected digits: A variable number of digits collected.

3.1.43 connection: An association of transmission channels or circuits, switching and other functional units set up to provide a means for a transfer of information between two or more points in a telecommunications network.

3.1.44 connection control: The set of functions used for setting up, maintaining and releasing a communication path between two or more users or a user and a network entity, e.g., a dual tone multifrequency receiver.

3.1.45 consumer: Refer to ITU-T Recommendation X.880.

3.1.46 control window: An interval during which an entity involved in call/service processing is subject to the control of the service control function.

3.1.47 customized dialing plan (CDP): Also known as “customer numbering plan” or “private numbering plan”, see “Access Code”.

3.1.48 database: An entity that stores information.

3.1.49 data management: Establishing, updating, and administering databases in the network.

3.1.50 data object: An individually addressable unit of information specified in a data template.

3.1.51 data template: A specified logical structure for a collection of data objects, including allowable ranges for their values and other data consistency specifications.

3.1.52 destination routing address: a list of *Called Party Number*s (primary and alternative).

3.1.53 destination user: The entity to which calls are directed.

3.1.54 detection point (DP): A point in basic call processing at which a processing event may be reported to the service control function and transfer of processing control can occur.

3.1.55 dialed digits: Untranslated address information collected/received from the originating line/trunk.

3.1.56 dialog: A conversation or an exchange of information.

3.1.57 digital subscriber signaling No. 1 protocol (DSS1): ITU-T Recommendation Q.931 Signaling specification for frame mode basic call control.

3.1.58 directory: Entry information and operation information held by a system. Refer to ITU-T Recommendation X.500 (clause 12).

3.1.59 directory access protocol (DAP): A directory protocol that defines the exchange of requests and outcomes between user and system. Refer to ITU-T Recommendation X.500 (clause 12).

3.1.60 directory entry (DE): Directory information consisting of a set of attributes. Refer to ITU-T Recommendation X.501 (clause 7.1).

3.1.61 directory information tree (DIT): Information in a directory arranged in a tree identifying different objects. Refer to ITU-T Recommendation X.501 (clause 7.1).

3.1.62 directory user agent: Directory information user. Refer to ITU-T Recommendation X.500 (clause 6.1).

3.1.63 distinguished name (DN): Directory name identifying one name of an object entry. Refer to ITU-T Recommendation X.501 (clause 9.4).

3.1.64 distributed functional plane (DFP): The plane in the intelligent network conceptual model containing functional entities and their relationships.

3.1.65 dual tone multi-frequency (DTMF): A push-button dialing method used to direct inband signaling entities.

3.1.66 dynamic arming/disarming: Enabling/disabling of a detection point by a service control function in the course of service control execution for a particular call/service attempt.

3.1.67 dynamic data: Information subject to change as a result of call/service processing.

3.1.68 elementary function: A primary or basic function that cannot be further decomposed.

3.1.69 entity: A part, device, sub-system, functional unit, equipment, or system that can be individually considered. In ISDN, the term is used to refer to a particular system or sub-system such as a user terminal or a digital exchange. It is also used to refer to a set of functions of a particular system at a location, e.g., the layer 2 functions of a signaling system at a user terminal.

3.1.70 event: A specific input to and/or output from a given state in a finite state machine model that causes a transition from one state to another.

3.1.71 event detection point (EDP): A detection point that is dynamically armed.

3.1.72 facility code: Code(s) used to select/activate a facility related service, e.g., a two-digit code preceded by “#” to select a preferred routing.

3.1.73 facility group: Indicates the particular group of facilities to route the call.

3.1.74 facility group member: Indicates the specific member of a trunk group or a multi-line hunt group.

3.1.75 feature: A reusable capability provided to a user by one or more services in a network.

3.1.76 feature code: Code(s) used to select/activate a service feature (e.g., forwarding, using two- or three-digit codes preceded by * or 11 or #, and which may precede subsequent digit selection).

3.1.77 feature interaction: A situation that occurs when an action of one feature affects an action or capability of another.

3.1.78 feature interactions manager: The entity in the SSF that provides mechanisms to support multiple concurrent instances of IN service logic instances and non-IN service logic instances on a single call.

3.1.79 finite state machine (FSM): A system having a finite number of states and specified transitions between states.

3.1.80 finite state machine model: An operational model of an entity that is described by the finite set of states the entity can be in and the finite set of transitions possible from one state to another.

3.1.81 functional entity (FE): An entity that comprises a specific set of functions at a given location.

3.1.82 functional entity [in telecommunications service provision applications]: A grouping of service providing functions in a single location and a subset of the total set of functions required to provide the service.

3.1.83 functional entity action (FEA): An action performed by a functional entity as a result of a specific stimulus while the functional entity is in a specific state.

3.1.84 functional routine: Logic that controls the performance of a set of actions to accomplish “routine” tasks, e.g., retrieve information, pass information, etc.

3.1.85 global functional plane (GFP): The plane in the intelligent network conceptual model that defines service independent building blocks (SIBs) used in providing service features.

3.1.86 inband (inband signaling): A signaling method in which signals are sent over the same transmission channel or circuit as the user’s communication and in the same frequency band as that provided for the users - an example is DTMF.

3.1.87 information element: An element in the information flow.

3.1.88 information flow: An interaction between a communicating pair of functional entities.

3.1.89 intelligent network (IN): A telecommunications network architecture that provides flexibility for facilitating the introduction of new capabilities and services, including those under customer control.

3.1.90 intelligent network application protocol (INAP): A protocol for intelligent network applications contained in layer 7 (application of the OSI model).

3.1.91 IN conceptual model (INCM): A planning model used for defining the intelligent network architecture.

3.1.92 IN supported service: A service provided using the capabilities of the intelligent network.

3.1.93 IN switching state model (IN-SSM): Provides an object-orientated finite state machine description of SSF/CCF IN call/connection processing in terms of IN call/connection states.

3.1.94 intelligent peripheral (IP): A physical entity that implements the intelligent network specialized resource function.

3.1.95 Interface: A shared boundary, for example, the boundary between two sub-systems or two devices.

3.1.96 ISDN User Part (ISUP): The signaling system No. 7 protocol that provides the signaling functions required to support basic bearer services and supplementary services for voice and non-voice applications in an integrated services digital network. Refer to T1.113.

3.1.97 layer: A conceptual region that embodies one or more functions between an upper and a lower logical boundary within a hierarchy of functions.

3.1.98 leg: A representation within a call processing state model representing a telecommunication path towards some addressable entity (e.g. a path toward a user, intelligent peripheral unit etc.).

3.1.99 library: An assembly of objects, routines, programs, etc. that may be drawn upon for use in the performance of functions.

3.1.100 local exchange (LE) [local central office]: An exchange in which subscriber lines terminate.

3.1.101 management function: A set of processes used for the management of an entity (e.g., data base management capabilities covering maintenance of operations, administration, maintenance, and provisioning).

3.1.102 manager: A function that directs and/or controls operations of a function or an assembly of functions to allow a functional entity to perform all or a part of the expected functional entity actions.

3.1.103 monitor window: An interval during which an entity performs the monitoring function at the direction of a service control function.

3.1.104 multiple association control function (MACF): Represents the rules and regulations governing the coordination of set of peer-to-peer communications within an application entity invocation (AEI).

3.1.105 network access point (NAP): A physical entity that provides network access for users. It contains the call control agent function and may include the call control function.

3.1.106 network data: Data that are specific to the functionality of the network.

3.1.107 network interworking: The cooperation of networks to process, manage, and create services, which span multiple networks.

3.1.108 network manager (NM): The Network Manager is the entity providing the basic switching and transmission capabilities as well as the service execution capabilities (offered by the SCP, SDP and IP) to the Service Manager. The NM is also responsible for the development and maintenance of the transmission, switching, and service execution capabilities. The NM encompasses both the TMN network management functionality and the network element management functionality.

3.1.109 network operator: The network operator is responsible for the development, provision, and maintenance of real-time networking services and for operating the corresponding networks.

3.1.110 network provider: The organization that maintains and operates the network components required for IN functionality. A network provider may also take more than one role, e.g. also acting as Service Provider.

3.1.111 object: An intrinsic component of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions.

3.1.112 open systems interconnection (OSI): The concept of interconnecting systems in accordance with the architecture described in the Open System Interconnection Reference model. Refer to ITU-T Recommendation X.200.

3.1.113 operator services information: Information sent between operator services entities identifying charging and service type options.

3.1.114 optional feature: A service feature added to core features to optionally enhance a service offering.

3.1.115 originating line information (OLI): Information indicating a toll class of service for the call.

3.1.116 persistent data: Information that endures beyond a single instance of use, e.g., longer than one call attempt.

3.1.117 personal mobility (T1.702): The ability of a user to access telecommunication services at any terminal on the basis of a personal identifier, and the capability of the network to provide those services according to the user's service profile. Personal mobility involves the network capability to locate the terminal associated with the user for the purposes of addressing, routing, and charging of the user's call.

3.1.118 physical plane: The plane in the intelligent network conceptual model containing elements and their interfaces that implement functional entities.

3.1.119 plain old telephone service (POTS): A call that requires nothing more than basic call handling.

3.1.120 plane: A part of the intelligent network conceptual model.

3.1.121 point in call (PIC): A state in a basic call state model.

3.1.122 prefix: Any prefix digits that need to be input by the calling party, e.g. 0, 1.

3.1.123 primary rate access (PRI): A user-network access arrangement that corresponds to the primary rate of 1544 kbit/s. The bit rate of the D-channel for this type of access is 64 kbit/s.

3.1.124 population rules: Condition(s) and respective values for inclusion, e.g. semantics to enable formation of operations sent by the SSF to the SCF using particular information elements.

3.1.125 public switched telephony network (PSTN): A telecommunications network set up to perform telephonic services for the public subscribers.

3.1.126 redirection information: Refer to T1.113.

3.1.127 relationship: The complete set of information flows, where they exist, between two functional entities.

3.1.128 relative distinguished name (RDN): An object or alias entry consisting of a set of attribute type and value pairs. Refer to ITU-T Recommendation X.501 (clause 9.3).

3.1.129 resource: In telecommunications, any network element that can be drawn upon in providing service, e.g. a circuit, a receiver, etc.

3.1.130 route index: A pointer to a specific trunk group.

3.1.131 route list: A list of trunk groups or a “route index” (if the call does not terminate on this SSF/CCF).

3.1.132 service: That which is offered to customers in order to satisfy a telecommunication requirement.

3.1.133 service address information: Information that represents the result of trigger analysis and allows the SCF to choose the appropriate service logic.

3.1.134 service control: Direction of the functions or processes used to provide a specific telecommunications service.

3.1.135 service control customization: Functionality to personalize a stand-alone commercial offering, by the server on behalf of a client.

3.1.136 service control function (SCF): The application of service logic to control functional entities in providing intelligent network services.

3.1.137 service control function identifier (SCFID): Indicates the SCF and enables the assisting SSF to identify to which SCF the assist request instructions should be sent.

3.1.138 service control point (SCP): An entity in the intelligent network that implements a service control function.

3.1.139 service creation: An activity whereby the capability to provide a supplementary service is brought into being from specification to development and verification.

3.1.140 service creation environment function (SCEF): The set of functions to support service creation processing by altering service logic and service data on behalf of the controlling node.

3.1.141 service creation platform: A set of service independent objects or functions that allow the creation of services in an intelligent network.

3.1.142 service creation process: The conception, design, and implementation of a capability to provide a service.

3.1.143 service data: Customer and/or network information required for the proper functioning of a service.

3.1.144 service data function (SDF): The set of functions that provides for the management of service data in accordance with a service data template.

3.1.145 service data point (SDP): A physical entity that implements a service data function.

3.1.146 service data template: A data template related to a specific service logic program.

3.1.147 service feature (SF): A reusable part of one or more service capabilities forming all or part of a service.

3.1.148 service independence: Not necessarily specific to one service.

3.1.149 service logic (SL): A sequence of processes/functions used to provide a specific service.

3.1.150 service logic processing program (SLP): A software program containing service logic.

3.1.151 service logic processing program (use) instance (SLPI): The invocation and application of a particular service logic program in providing a service or a service feature for a specific call/service attempt.

3.1.152 service management: Management of user and/or network information required for the proper operation of a service.

3.1.153 service management access function (SMAF): The entity equivalent to the workstation function between network operators and/or subscribers and network service management functional entities.

3.1.154 service management function (SMF): The set of processes that support the management of user and/or network information, including service data and service logic programs that are required for the proper operation of a service.

3.1.155 service management system (SMS): A set of service management functions.

3.1.156 service node (SN): A physical entity that contains the service control function, service data function, specialized resource function, and service switching/call control functions. The SSF/CCF is closely coupled to the SCF within the SN and is not accessible by other SCFs.

3.1.157 service plane: The plane in the intelligent network conceptual model that contains services, service entities, and their relationships.

3.1.158 service provider: An organization that commercially manages services offered to service subscribers. The network operator may be the service provider.

3.1.159 service subscriber (SS): An entity that contracts for services offered by Service Providers.

3.1.160 service switching and control point (SSCP): A physical entity that contains the service control function, service data function, and the service switching/call control functions.

3.1.161 service switching function (SSF): The set of processes that provide for interaction between a call control function and a service control function.

3.1.162 service switching management entity (SSME): Functionality of an entity controlling the system management of the node.

3.1.163 service switching point (SSP): A physical entity that implements a service switching function.

3.1.164 serving area ID: Identifies the local serving area where the network provider operates.

3.1.165 single association control function (SACF): Represents the rules and regulation governing the use of the ASEs that are being used for communication over a single Application Association to a peer.

3.1.166 single association object (SAO): The representation of the functions that are needed to communicate over a single Application Association to a peer.

3.1.167 single-ended service feature: A feature, e.g. call/service attempt manipulation that applies to only one of the parties that may be involved on a call/service attempt.

3.1.168 single point of control: A control relationship where the same phase or aspect of a call/service attempt is influenced by one and only one service control function.

3.1.169 specialized resource function (SRF): The set of functions that provide for the control and access to resources used in providing services in the intelligent network.

3.1.170 state (in FSM): A description of an entity defined by the values of its object attributes at a given point in time.

3.1.171 state (in SDL): A condition in which the action of a process is suspended awaiting an input.

3.1.172 static arming/disarming: Enabling/disabling of a detection point, as directed by a service management function, to cause a specified action by call/service processing whenever a specific point in call/service processing is encountered.

3.1.173 static data: Information that remains unchanged for the duration of a call or incident of use of a service. (Usually controlled by a source external to the network.)

3.1.174 supplier: see ITU-T Recommendation X.880.

3.1.175 telecommunication management network (TMN): The entity that provides the means used to transport and process information related to management functions for the telecommunications network.

3.1.176 terminal mobility (T1.702): The ability of a terminal to access telecommunications services from different locations and while in motion, and the capability of the network to identify, locate and communicate with that terminal. Terminal mobility while not on a call may involve roaming. Or while on a call may involve handoff.

3.1.177 terminal type: Indicates the type of terminal or the originator (e.g., DTMF phone, ISDN terminal).

3.1.178 transaction capabilities (TC) (T1.114-2000): A means based upon the OSI Reference Model to support applications in telecommunications networks.

3.1.179 transaction capabilities application part (TCAP): The Application layer services and protocols at layer 7 of the OSI model consisting of the Component sub-layer and the Transaction sub-layer (T1.114).

3.1.180 transit network selection: This identifier if present identifies the Carrier Identification Code and the Circuit Code.

3.1.181 transition: In a finite state machine model, a change in the state of an entity resulting from a change in the values of its object attributes.

3.1.182 traveling class mark: Provides information for routing or screening and allows for carrying class of service information along with calling number through a network. An example of traveling class mark is the means to override facility restriction level as a call is routed through a network.

3.1.183 trigger: A stimulus for initiating an action.

3.1.184 trigger detection point (TDP): A detection point in basic call processing that is statically armed.

3.1.185 type of call: e.g. “local”, “national”, “international”, etc.

3.1.186 universal personal telecommunications (UPT): A telecommunications service that enables access to telecommunications services while allowing personal mobility.

3.1.187 unbind: A mechanism used during Association Control for authentication. Refer to ITU-T Recommendation X.500.

3.1.188 user: An entity external to the network that uses its service(s).

3.1.189 vendor or implementation independent: The characteristic that products from different vendors are able to work together in the same environment, and/or, physical units serving as the same functional entity(ies) produced by different vendors can be used interchangeably.

3.1.190 virtual private network (VPN): When existing, all the business groups, and/or PBXs on the host network that belong to the same customer private network.

3.2 Abbreviations

AC	Application Context
ACM	ISUP Address Complete Message
CAN	Application Context Negotiation

AD	Adjunct
AE	Application Entity
AEI	Application Entity Invocation
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASE	Application Service Element
BCP	Basic Call Process
BCM	Basic Call Model
BCSM	Basic Call State Model.
BGID	Business Group Identity
BRI	Basic Rate Interface
CAC	Carrier Access Code
CCA	Call Control Agent
CCAF	Call Control Agent Function
CCF	Call Control Function
CCPN	Call Completion to Portable Number
CDP	Customized Dialing Plan
CM	Call Manager
COT	ISUP Continuity message
CPG	ISUP Call Progress message
CS	Capability Set
DAP	Directory Access Protocol
DFP	Distributed Functional Plane
DN	Directory Number
DN	Distinguished Name
DP	Detection Point
DSA	Directory System Agent
DSS1	Digital Subscriber Signaling No. 1 Protocol
DTMF	Dual Tone Multi Frequency
DUA	Directory User Agent
EAMF	Equal Access Multi-Frequency
EDP	Event Detection Point
EDP-N	Event Detection Point-Notification
EDP-R	Event Detection Point-Request
FCI	Furnish Charging Information
FEA	Functional Entity Action
FEAM	Functional Entity Access Manager

FIM	Feature Interactions Manager
FSM	Finite State Machine
GFP	Global Functional Plane
IAM	Initial Address Message
IEC	International Electrotechnical Commission
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
INCM	IN Conceptual Model
IN-SM	IN Switching Manager
IN-SSM	IN Switching State Model
IP	Intelligent Peripheral
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISUP	Integrated Services Digital Network-User Part.
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LE	Local Exchange
MACF	Multiple Association Control Function
MF	Multi Frequency
NAP	Network Access Point
NCA	Non-Call Associated
NEF	Network Element Function
NM	Network Manager
NSAP	Network Service Access Point
OFC	Off-line Charging (billing/accounting information)
OLE	Originating Local Exchange
OLI	Originating Line Information
ONC	Online Charging (user access information)
OSF	Operator System Function
OSI	Open Systems Interconnection
PCS	Personal Communication Service
PIC	Point In Call
PRI	Primary Rate Interface
PSTN	Public Switched Telephony Network
RACF	Radio Access Control Function
RCF	Radio Control Function
RDN	Relative Distinguished Name
REG	Registration

RLF	Radio Link Function
ROS	Remote Operations
ROSE	Remote Operations Service Element
SACF	Single Association Control Function
SAO	Single Association Object
SCEF	Service Creation Environment Function
SCF	Service Control Function
SCF FSM	Service Control Function Finite State Machine
SCFID	Service Control Function Identifier
SCI	Send Charging Information
SCME	Service Control Function Management Entity
SCME FSM	Service Control Function Management Entity Finite State Machine
SCP	Service Control Point
SCSM	Service Control Function Call State Model
SDF	Service Data Function
SDF FSM	Service Data Function Finite State Machine
SDL	Specification and Description Language
SDME	Service Data Function Management Entity
SDP	Service Data Point
SDSM	Service Data Function Call State Model
SF	Service Feature
SIB	Service Independent Building Block
SL	Service Logic
SLEE	Service Logic Execution Environment
SLCP	Service Logic Control Program
SLMP	Service Logic Management Program
SLP	Service Logic Processing Program
SLPI	Service Logic Processing Program Instance
SM	Service Manager
SMF	Service Management Function
SMP	Service Management Point
SMS	Service Management System
SN	Service Node
SQL	Structured (database) Query Language
SRF	Specialized Resource Function
SS7	Signaling System No. 7
SSCP	Service Switching and Control Point

SSF	Service Switching Function
SSF FSM	Service Switching Function Finite State Machine
SSME	Service Switching Management Entity
SSME FSM	Service Switching Management Entity - Finite State Machine
SSP	Service Switching Point
TC	Transaction Capabilities
TCAP	Transaction Capabilities Application Part
TDP	Trigger Detection Point
TDP-N	Trigger Detection Point- Notification
TDP-R	Trigger Detection Point-Request
TMN	Telecommunication Management Network
UPT	Universal Personal Telecommunication
VPN	Virtual Private Network

4 Overview

This overview provides a brief description of the standard, IN Conceptual Model (INCM), IN internetworking, and IN support for wireless services.

4.1 Introduction

This standard has been developed to support the unique needs of the North American telecommunications environment. The T1 IN standard (version 1) (this standard) is based upon the ITU-T IN CS-1 Recommendations (1995) plus the additional protocol operations needed to support the North American requirements of T1 IN.

The ITU-T has specified the IN Conceptual Model (INCM) consisting of four planes, namely:

- Service Plane (SVP)
- Global Functional Plane (GFP)
- Distributed Functional Plane (DFP)
- Physical Plane (PHP).

Of these, the T1 IN standard incorporates the Distributed Functional Plane and Physical Plane. The T1 IN does not contain aspects of the Service Plane or Global Functional Plane. The INCM is described in 4.2. Full details of the INCM are presented in ITU-T Recommendation Q.1201.

4.2 IN Conceptual Model (INCM)

The INCM consists of four “planes” where each plane represents a different abstract view of the capabilities provided by an IN-structured network. These views address service aspects, global functionality, distributed functionality, and physical aspects of an IN. The IN Conceptual Model should not be considered in itself an architecture. It is a framework for the design and description of the IN architecture.

4.2.1 Service plane

The service plane represents an exclusively service-oriented view. This view contains no information whatsoever regarding the implementation of the services in the network, e.g., an “IN-type” implementation is not visible. All that is perceived is the network’s service-related behavior as seen, for example, by a service user. Services are composed of one or more Service Features (SFs), which are the “lowest level” of services.

4.2.2 Global functional plane

The global functional plane (GFP) models an IN-structured network as a single entity. Contained in this view is a global (network-wide) basic call processing (BCP) service independent building block (SIB), other service independent building blocks (SIBs), and point of initiation (POI) and point of return (POR) between the BCP and a chain of SIBs.

4.2.3 Distributed functional plane

The distributed functional plane (DFP) models a distributed view of an IN-structured network. Each functional entity (FE) described in the DFP may perform a variety of functional entity actions (FEAs). Any given FEA may be performed within different functional entities. However, a given FEA may not be distributed across functional entities. Within each functional entity, various FEAs may be performed by one or more elementary functions. Service-independent building blocks (SIBs) are realized in the distributed functional plane (DFP) by a sequence of particular FEAs performed in the functional entities. Some of these FEAs result in information flows between functional entities.

4.2.4 Physical plane

The Physical Plane models the physical aspects of IN-structured networks. The model identifies the different physical entities and protocols that may exist in real IN-structured networks. It also indicates which functional entities are implemented in which physical entities.

4.3 IN internetworking

IN internetworking is a process in which several networks cooperate to provide a service. The need for internetworking capabilities arises when customers wish to access the services in other networks or that cannot be provided by one network alone. An example of such a situation is when the data needed by a service (e.g., UPT or VPN) reside in a network that is different from the one in which the call has originated.

Although the involved networks may have different access types (i.e., PSTN, ISDN, etc.), as well as different levels of IN structure, the services should be provided to customers in a consistent way, regardless of such differences.

Clause 5 describes the functional relationship between IN functions that are located in two different networks, namely, SCF-SDF relationship. The following observations can be made:

- internetworking between the SSF in one network and the SCF in another one is not a requirement;
- internetworking between the SRF in one network and the SCF in another one is not a requirement;
- for several IN-supported services (e.g., UPT and VPN), the SCF may perform translation and validation via information exchange with the SDF. The SCF-SDF functional relationship supports internetworking of IN structured networks.

The SCF-SSF and SCF-SRF intranetwork interfaces are not applicable to internetworking between IN structured networks in this version of the T1 IN standard. Refer to Annex C for a description of the SCF-SDF interface.

4.4 IN support for UPT

This standard is intended to support the UPT Service Set 1 capabilities, as defined in T1.701.

4.5 IN support for Personal Communication Service

This standard includes functionality to support terminal mobility, along with the personal mobility of UPT, in order to support initial Personal Communication Service offerings. The personal mobility service features should also be available to a mobile user in a wireless environment.

5 Functional architecture

5.1 Overview

IN call/service logic, processing encompasses call and connection processing in the SSF/CCF, service logic execution in the SCF, and the use of supporting resources and data in the SRF and SDF, respectively. This clause describes this IN call/service logic processing in terms of call modeling and modeling of service logic processing.

- Call modeling provides a high-level service and vendor/implementation independent abstraction of IN call and connection processing in the SSF and CCF. This abstraction provides an observable view of SSF/CCF activities and resources to the SCF, enabling the SCF to interact with the SSF in the course of executing service logic.
- The modeling of service logic processing provides an abstraction of SCF activities and resources needed to support this service logic execution, as well as an abstraction of SRF and SDF activities and resources accessible to the SCF.

Since this modeling only provides an observable (i.e., external) view of SSF/CCF, SCF, SRF, and SDF activities and resources, this modeling does not imply an obligation to vendors to implement functional entities into products as a one-to-one mapping of functional entity model components.

5.2 IN distributed functional plane

Figure 1 identifies the IN DFP model for T1 IN. This diagram depicts the functional entities and relationships applicable to T1 IN.

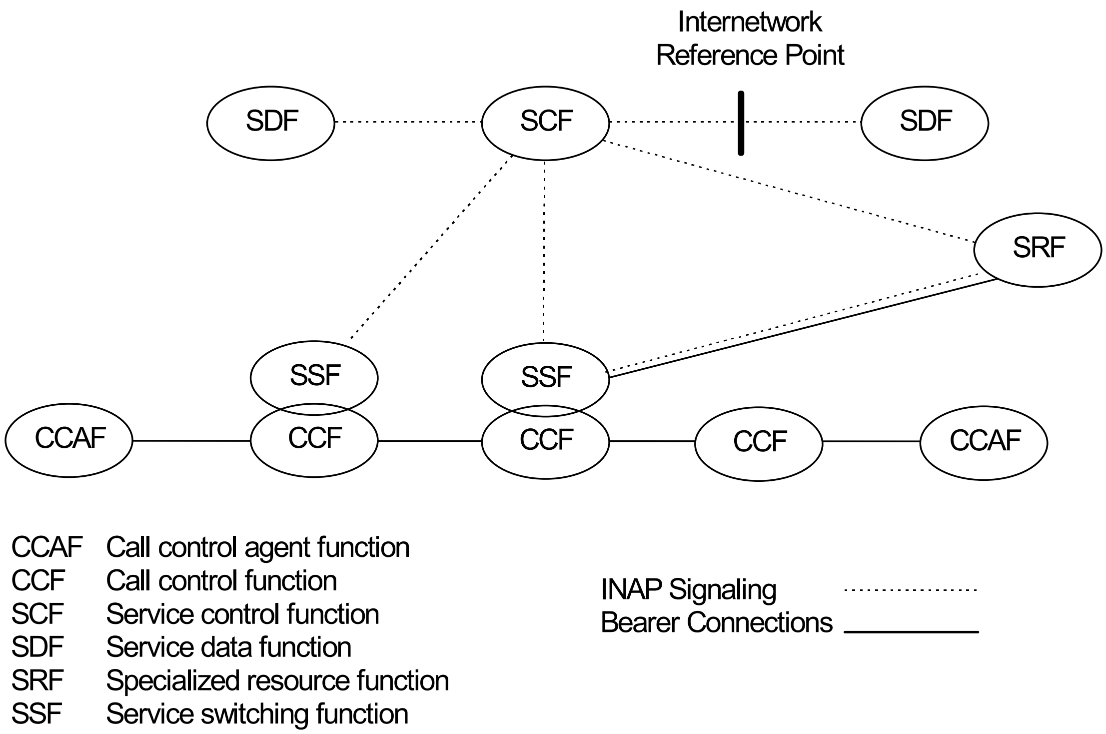


Figure 1 - IN distributed functional plane model for T1 IN

The IN DFP for this standard, as represented in Figure 1, has the following characteristics:

- only the CCAF, CCF, SSF, SCF, SDF, and SRF functional entities are included;
- only the relationships related to IN service execution are addressed, as shown in Figure 1;
- service management and administration aspects of each functional entity are implied, but not specifically addressed in this standard.

The functional entities related to IN service execution are defined as follows:

The CCA function (CCAF): The CCAF is the call control agent (CCA) function that provides access for users. The CCAF provides functions to support the interface between users and network call control functions. The CCAF

- provides for user access, interacting with the user to establish, maintain, modify and release, as required, a call or instance of service;
- accesses the service-providing capabilities of the call control function (CCF), using service requests (e.g., setup, transfer, hold, etc.) for the establishment, manipulation and release of a call or instance of service;
- receives indications relating to the call or service from the CCF and relays them to the user as required;

- maintains call/service state information as perceived by this functional entity.

The CC function (CCF): The CCF is the call control (CC) function in the network that provides call/service processing and control. The CCF

- establishes, manipulates and releases calls/connections as “requested” by the CCAF;
- provides the capability to associate and relate CCAF functional entities that are involved in a particular call and/or connection instance (that may be due to SSF requests);
- manages the relationship between CCAF functional entities involved in a call (e.g., supervises the overall perspective of the call and/or connection instance);
- provides trigger mechanisms to access IN functionality (e.g., passes events to the SSF).

The SS function (SSF): The SSF is the service switching (SS) function, which, associated with the CCF, provides the set of functions required for interaction between the CCF and a service control function (SCF). The SSF

- extends the logic of the CCF to include recognition of service control triggers and to interact with the SCF;
- manages signaling between the CCF and the SCF;
- modifies call/connection processing functions (in the CCF) as required to process requests for IN provided service usage under the control of the SCF.

The SC function (SCF): The SCF is the service control (SC) function that commands call control functions in the processing of IN provided and/or custom service requests. The SCF may interact with other functional entities to access additional logic or to obtain information (service or user data) required to process a call/service logic instance. The SCF

- interfaces and interacts with the service switching function/call control function, specialized resource function (SRF) and service data function (SDF) functional entities;
- contains the logic and processing capability required to handle IN provided service attempts.

The SD function (SDF): The SDF is the service data (SD) function that contains customer and network data for real time access by the SCF in the execution of an IN provided service. The SDF interfaces and interacts with SCFs as required.

NOTE - The SDF contains data relating directly to the provision or operation of IN-provided services. Thus it does not necessarily encompass data provided by a third party such as credit information, but may provide access to these data.

The SR function (SRF): The SRF is the specialized resource (SR) function that provides the specialized resources required for the execution of IN provided services (e.g., digit receivers, announcements, conference bridges, etc.). The SRF

- interfaces and interacts with SCF and SSF (and with the CCF);
- may contain the logic and processing capability to receive/send and convert information received from users;
- may contain functionality similar to the CCF to manage bearer connections to the specialized resources.

5.3 SSF/CCF model

5.3.1 Basic Call State Model (BCSM)

The BCSM provides a high-level model description of CCF activities required to establish and maintain communication paths for users. As such, the BCSM identifies a set of basic call and connection activities in a CCF and shows how these activities are joined together to process a basic call and connection (i.e., establish and maintain a communication path for a user).

Many aspects of the BCSM are not externally visible to IN service logic instances. The BCSM is primarily an explanatory tool for providing a representation of CCF activities that can be analyzed to determine which aspects of the BCSM will be visible to IN service logic instances, if any, and what level of abstraction and granularity is appropriate for this visibility.

The BCSM identifies points in basic call and connection processing when IN service logic instances are permitted to interact with basic call and connection control capabilities. In particular, the BCSM provides a framework for describing basic call and connection events that can lead to the invocation of IN service logic instances or should be reported to active IN service logic instances, for describing those points in call and connection processing at which these events are detected, and for describing those points in call and connection processing when the transfer of control can occur.

Figure 2 shows the components that have been identified to describe a BCSM, including: points in call (PICs), detection points (DPs), BCSM transitions, and events. PICs identify CCF activities associated with one or more basic call/connection states of interest to IN service logic instances. DPs indicate states in basic call and connection processing at which transfer of control from non-IN to IN service logic can occur. Other states may exist (e.g. within PICs) where call processing is suspended but IN service logic cannot be invoked. BCSM transitions indicate the normal flow of basic call/connection processing from one PIC to another. Entry events cause BCSM transitions into PICs. Exit events represent the result of PIC processing.

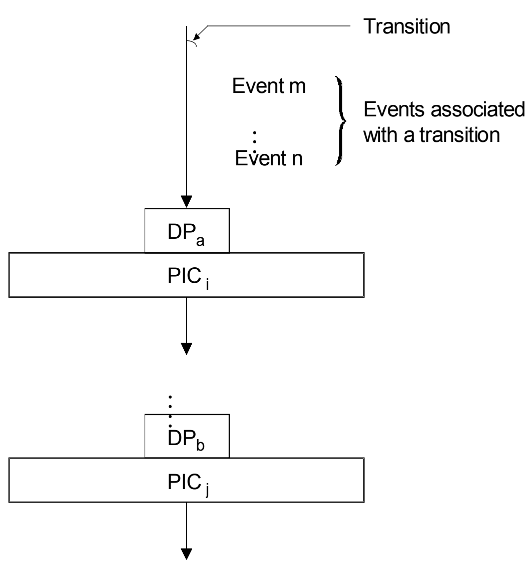


Figure 2 - BCSM components

The BCSM for T1 IN models existing switch processing of basic two-party calls, and reflects the functional separation between the originating and terminating portions of calls reflected in Figure 3 and Figure 4. In addition, though CCAF functionality is not explicitly modeled in the BCSM, a mapping is required between access signaling events and BCSM events, for each access arrangement supported by T1 IN.

Since the BCSM is generic, it may describe events that do not apply to certain access arrangements. It is important to understand and describe how each access arrangement applies to the BCSM.

5.3.2 T1 IN BCSM description

The T1 IN BCSM reflects the functional separation between the originating and terminating portions of calls as illustrated in Figure 3 and Figure 4. These figures show an originating half BCSM and a terminating half BCSM, each of which is managed by a functionally separate Basic Call Manager (BCM) in the SSF/CCF. The description is a starting point to identify the aspects of the BCSM that are visible to IN service logic instances, and the nature of the operations between the SSF/CCF and SCF (see clause 7).

NOTE - It is important to understand that the BCSMs shown in Figure 3 and Figure 4 do not depict all allowable call processing transitions that may result from SCF interaction with a call. A complete set of transitions is found in Table 1 (Originating BCSM) and Table 2 (Terminating BCSM).

In the following descriptions, the PICs are related at a high level to T1.607 ISDN call states. This is not intended to be a detailed formal definition of the relation between the PICs and T1.607 ISDN call states, but is intended as a point of reference to use in understanding the PICs. In particular, there are a number of possible ways in which the T1.607 call states may be traversed in certain situations that are not considered below.

To enable independence between services offered during one call session when the PICs may be traversed several times, it is necessary - at each PIC - to maintain available a specific set of data until the calling (controlling) user releases and to ensure that software resources are returned to a coherent status when call processing passes through the PICs.

For each PIC, an initial list of BCSM information that must be maintained, if available, is given. Information that is available at all PICs is given at the beginning of the O- and T- BCSM descriptions.

The information that is sent to the SCF at a given trigger detection point is a subset of the information described here. Other information may be available at a given PIC that is not used by processing at the PIC or is only used by underlying call processing.

In order to maintain uniqueness of DP names between the originating and terminating half BCSMs, “O” and “T” are prefixed to certain originating and terminating DP names, respectively.

For ease of reference, the DPs associated with the transition implied by each entry and exit event for each PIC are listed along with the PIC descriptions.

5.3.2.1 Originating BCSM for T1 IN

The originating half of the BCSM corresponds to that portion of the BCSM associated with the originating party (see Figure 3).

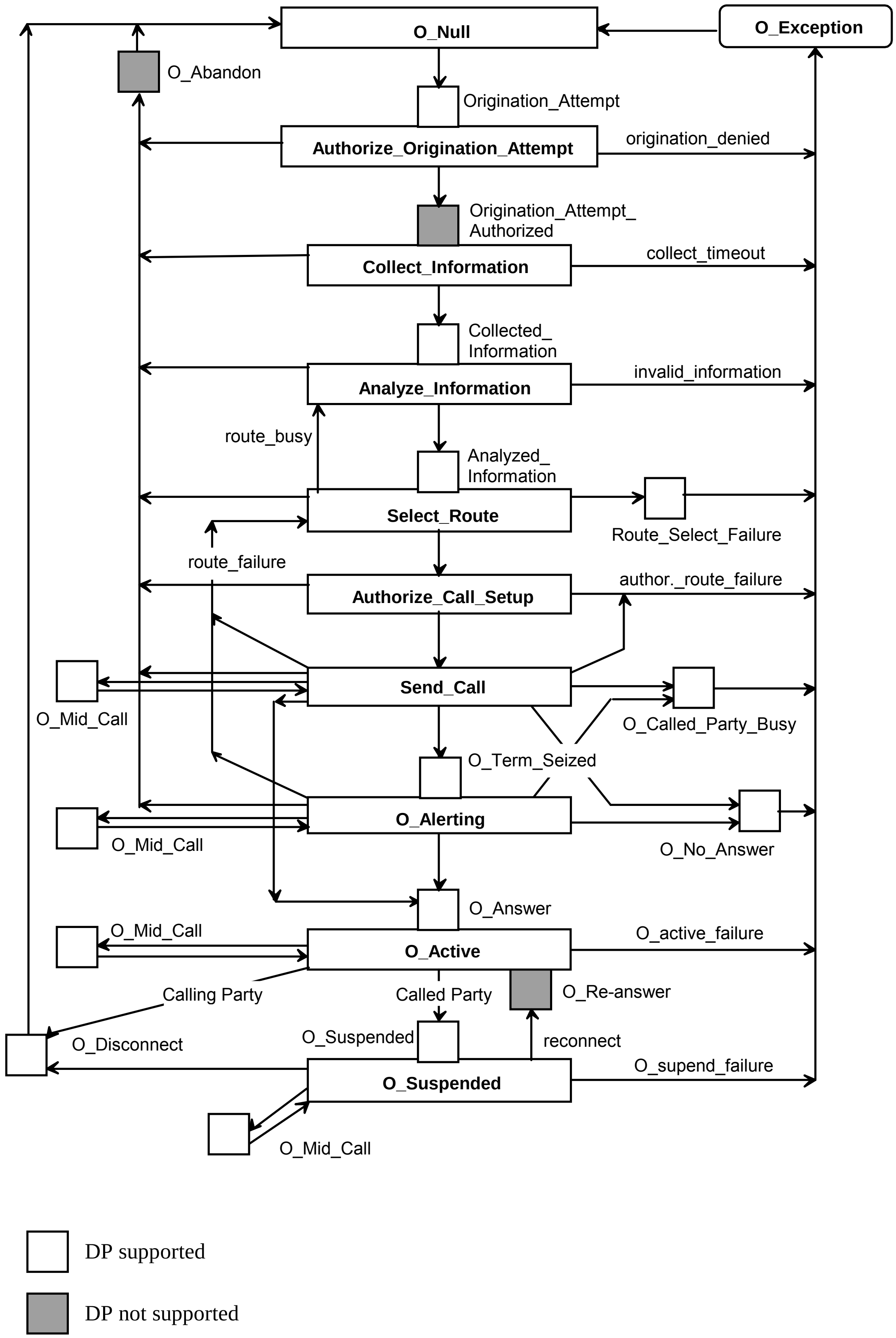


Figure 3 - Originating BCSM for T1 IN

The following information is available at all PICs in the O-BCSM:

- Service Address Information - see clause 3.
- Calling Party Category - see T1.113 Calling Party's Category signaling information.
- Call Gapping Encountered - indicates the code/call control in effect at the SSF.
- Location Number - see ITU-T Recommendation Q.763. Provided if the calling party is a mobile subscriber.
- Original Called Party ID - see T1.113 Original Called Number parameter.
- Redirecting Party ID - see T1.113 Redirecting Number parameter.
- Redirection Information - see T1.113 Redirection Information parameter.
- Additional Calling Party Number - see T1.113 Generic Address parameter.
- ISDN Access Related Information: - see T1.113 Access Transport Parameter.

The description of each of the PICs in the originating half of the BCSM follows.

5.3.2.1.1 O_Null

<u>Entry event:</u>

Disconnect and clearing of a previous call (O_Disconnect DP or O_Abandon event), or default handling of exceptions by SSF/CCF completed.

<u>Functions:</u>

Interface (line/trunk) is idled (no call exists, no call reference exists, etc.). Supervision is being provided.

<u>Information Available :</u>

After detecting the Origination_Attempt event, the SSF/CCF has the following information available associated with the originating call portion, with restrictions as noted. If the SSF/CCF determines that the origination is denied, the cause of the failed authorization is also known.

A. Bearer Capability - see T1.113 User Service Information and T1.607 Bearer Capability information element.

B. Calling Party Number - see T1.113 Calling Party Number signaling information. This information is available at the SSF/CCF for a non-ISDN line and may be available for SS7 trunks, but is not available from trunks supported by conventional signaling or private-facility trunks. For a DSS1 interface, this is determined by the information provided in the SETUP message or by the default number assigned to the caller (see ISDN SETUP information below).

C. Service Profile Identifier (SPID) - see T1.610. This information may be available at the SSF/CCF if the calling party is served by a BRI interface on this SSF/CCF.

D. *Called Party Number* - see T1.113 *Called Party Number* signaling information. Used to identify the called party in the forward direction. Available only for trunks or ISDN lines.

E. Charge Number - see clause 3. This information is available at the SSF/CCF for a non-ISDN line or ISDN interface served by the SSF/CCF and may be available for trunks supported by SS7, but is not available for trunks supported by conventional signaling or private-facility trunks.

F. Class of Service - see clause 3.

G. Calling Party Business Group ID (BGID) - see clause 3 Business Group ID. This information is available for a non-ISDN line, ISDN interface, private-facility trunk group, or possibly an SS7 trunk when the caller is a member of a Business Group.

H. Calling Facility Group - see clause 3, facility group. Available on conventional or SS7 trunks.

I. Calling Facility Group Member - see clause 3, facility group member. Available on conventional or SS7 trunks.

J. Traveling Class Mark - see clause 3.

K. Feature Code - see clause 3. Available, if used, for a party served by an ISDN interface or for an SS7 trunk.

L. Access Code - see clause 3. Available, if used, for a party served by an ISDN interface.

M. Operator Services Information - see clause 3. This information element is not included in a SETUP message containing the keypad information element.

N. ISDN SETUP feature-related information - see T1.607. The SSF/CCF receives a SETUP message from a DSS1 interface and this SETUP message can contain the following information:

- Bearer Capability - see T1.607 Bearer Capability information element.
- Progress Indicator - see T1.607 Progress Indicator information element.
- Keypad Facility - see T1.607 Keypad Facility information element. This information element is not expected in a SETUP message also containing the *Called Party Number*, transit network selection, or Operator Services Information information elements.
- Feature activation - see T1.610 Feature Activation information element.
- Calling party number - see T1.607 Calling Party Number information element.
- *Called Party Number* - see T1.607 *Called Party Number* information element.

  The *Called Party Number* information element is sent when en bloc sending is used and the keypad information element is not present. When the type of number and numbering plan identification field within the *Called Party Number* information element is set to “unknown,” the SSF/CCF treats the string as if it has been received within a keypad information element. In this case, it is not expected to be sent with the transit network selection or Operator Services Information information elements.
- Calling party subaddress and called party subaddress - see T1.607 Calling Party Subaddress and Called Party Subaddress information elements.
- Transit Network Selection - see T1.607 Transit Network Selection information element. This information element is included in a SETUP message containing other information elements than the Keypad information.
- Facility Information - see T1.610 Facility information element. This element may identify User Service Information or Facility Information.
- Other information, as defined by T1.610. Some of this information may be of interest to the SCF.

O. ISDN User Part IAM feature related information. The IAM can contain the following information (see T1.113):

- Nature of connection indicators- see T1.113 Nature of Connection Indicators parameter.
- Forward call indicators - see T1.113 Forward Call Indicators parameter. The caller's access is identified as ISDN or non-ISDN, and an indication is given of whether an end-to-end SS7 supported connection is required.

- User service information - see T1.113 User Service Information parameter. For the purposes of T1 IN, this parameter identifies the call as circuit-mode/speech, circuit-mode/3.1-kHz audio, circuit-mode/unrestricted digital information (64 kbit/s), or circuit-mode/restricted digital information.
- *Called Party Number* - see T1.113 *Called Party Number* parameter.
- Calling party number - see T1.113 Calling Party Number parameter.
- Generic Address - see T1.113 Generic Address parameter. More than one generic address parameter may be present within a given IAM.
- Generic name - see T1.113 Generic Name parameter.
- Charge number - see clause 3.
- Transit network selection - see T1.113 Transit Network Selection parameter. This parameter, if present, identifies the Carrier Identification Code and the Circuit Code.
- Carrier selection - see clause 3.
- Feature Code - see clause 3.
- Generic digits - see T1.113 Generic Digits parameter. May contain a traveling class mark (network operator-specific).
- Other parameters may be included in the IAM. These parameters may be included because of features provided by other switches in the connection (e.g., information relating to the call being forwarded). Some of this information may be of interest to the SCF.

Any information relating to switch-based features that have already been invoked for the call will also be available.

Exit event:

A. Indication of desire to place outgoing call (e.g., off hook, SETUP message, ISUP IAM message). (DP: Origination_Attempt).

B. The following exception exit events are applicable to the O_Null PIC. For this PIC, if the call encounters one of these exceptions during O_Null PIC processing, the exception event is not visible because there is no corresponding DP.

- The O_Abandon event occurs when the calling party disconnects. For example, this event can result from one of the following:
- The SSF/CCF receives an on-hook indication from a caller served by a non-ISDN line, following switch hook flash timing;
- The SSF/CCF receives a call clearing message from a caller served by an ISDN interface;
- The SSF/CCF receives a disconnect indication from a conventional trunk or private facility trunk;
- SSF receives a Release Message from an SS7 trunk.

***Corresponding T1.607 call state:* O_Null**

5.3.2.1.2 Authorize_Origination_Attempt

Entry event:

An indication is available that the originating terminal needs to be authorized. (DP: Origination_Attempt).

Functions:

The originating terminal rights should be checked using the calling party's identity and service profile. The authority/ability of the party to place the call with given properties (e.g., bearer capability, line restrictions) is verified. The types of authorization to be performed may vary for different types of originating resources (e.g., for lines vs. trunks).

Information available:

1. After detecting the Origination_Attempt_Authorized event, the SSF has the same information available associated with the originating call portion as it did after detecting the Origination_Attempt event in the O_Null PIC.
2. If the SSF determines that the origination is denied, the cause of the failed authorization is also known.

Exit event:

A. An indication is received that the authorization is successful. The O_BCSM moves to the Collect_Information PIC.

B. A disconnection indication is received from the originating party.

C. An indication is received that the call origination is denied. The O_BCSM moves to the O_Exception PIC.

5.3.2.1.3 Collect_Information

Entry event:

Authority/ability to place outgoing call verified.

Functions:

A. Initial information package/dialing string (e.g., service codes, prefixes, dialed address digits) being collected from originating party. Information being examined according to dialing plan to determine end of collection. No further action may be required if an en bloc signaling method is in use (e.g., an ISDN user using en bloc signaling, an incoming SS No. 7 trunk).

B. The SSF/CCF shall be able to support subsequent digit collection according to trigger criteria assigned before sending the query. For example, if a feature code (e.g., *64) is entered, the SSF/CCF may:

- collect digits according to the normal dialing plan, or
- collect a variable number of digits.

Information Available:

After the SSF/CCF determines that information collection is complete, the SSF/CCF has the following information available associated with the originating call portion:

A. Charge Number, Calling Party Number, Calling Party BGID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, other feature-related information, Additional Calling Party Number, Facility Information. This information is available for each access type under the conditions defined in the O_Null PIC.

B. Collected Information - As described below.

From a non-ISDN line or DSS1 interface, the collected information consists of one or more of the following:

- Access Codes within a Customized Dialing Plan (CDP) - see clause 3.

  The Customized Dialing Plan (CDP) in force may specify that after a given access code is dialed, more digits are to be collected according to the “normal dialing plan,” i.e., the dialing plan in force. In this case, Access Code and Collected Address Information are known. If the CDP in force specifies that after a given access code is dialed, a variable number of digits are to be collected, then Access Code and Collected Digits are known.

- Feature Code - see clause 3.

  If the numbering plan in force specifies that after a given feature code is dialed, more digits are to be collected according to the “normal dialing plan”, then Feature Code and Collected Address Information are known. If the dialing plan in force specifies that after a given feature code is dialed, a variable number of digits are to be collected, then Feature Code and Collected Digits are known. The service associated with the feature code is dependent upon the user's service profile.

- Facility Code - see clause 3. This information may be provided if and when facility selective service signaling is supported.
- Feature Activation - see T1.610 Feature Activation information element. If the CDP in force specifies that after a given feature activator is received, more digits are to be collected according to the numbering plan, then Feature Activation Indicator and Collected Address Information are known. If the CDP in force specifies that after a given feature activator is received, a variable number of digits are to be collected, then Feature Activation Indicator and Collected Digits are known.
- Prefix - see clause 3.
- Carrier Access Code/Carrier Identification Code - see clause 3. The caller may dial a Carrier Access Code (CAC) (e.g., a 101XXXX for use on this call). When the caller is served by an ISDN interface, a Carrier Identification Code, i.e., XXX or XXXX, may be received by the SSF/CCF within the transit network selection information element of the ISDN SETUP message.
- Collected Address Information - see clause 3. Available as per the numbering plan.
- Numbering Plan Indicator - see T1.113 Numbering Plan Indicator signaling information.
- Collected Digits - see clause 3. The numbering plan in force may specify that after a given Feature Activation, Feature Code, or Access Code within a CDP is dialed, a variable number of digits are to be collected using normal inter-digit timing. In this case, these collected digits are also known at this time.

From a conventional trunk interface, the Collected Information consists of one or more of the following:

- Charge Number- see clause 3. This is only known from a conventional trunk when Equal Access Multi-Frequency (EAMF) signaling is used on the originating trunk (network operator-specific). In this case, the charge number is provided in the second stage of overlap outpulsing.
- Collected Address Information - as defined above for non-ISDN line or DSS1 interface
- Carrier Identification Code - see clause 3. This is known if EAMF signaling is used on the originating trunk (network operator-specific).
- Numbering Plan Indicator - see T1.113 Numbering Plan Indicator signaling information. The address received is expected to conform to ITU-T Recommendation E. 164.

- Prefix - as defined above for non-ISDN line or DSS1 interface.
- Carrier Selection - see clause 3. This information is only provided when EAMF signaling is used on the originating trunk (network operator-specific).
- Originating Line Information - see clause 3. This information is only known when EAMF signaling is used on the originating trunk (network operator-specific). In this case, the Originating Line Information is sent during the second stage of overlap outpulsing.

From an SS7 trunk interface, the Collected Information consists of the information provided in the ISDN User Part *Called Party Number* and transit network selection parameters, and possibly a Traveling Class Mark and other feature-related information as described above for contents of the ISDN User Part IAM feature-related information.

From a private-facility trunk, the collected information consists of one or more of the following:

- Access Code within a CDP - as defined above for a non-ISDN line
- Feature Code - as defined above for a non-ISDN line
- Facility Code - as defined above for a non-ISDN line
- Collected Address Information - as defined above for a non-ISDN line
- Numbering Plan Indicator- as defined above for a non-ISDN line
- Prefix - as defined above for a non-ISDN line
- Carrier Access Code - as defined above for a non-ISDN line
- Traveling Class Mark (see clause 3) - if provided in the generic digits parameter of the IAM
- Facility Restriction Level - see clause 3.

Exit events:

A. Availability of complete initial information package/dialing string from originating party. (This event may have already occurred in the case of en bloc signaling, in which case the waiting duration in this PIC is zero.) (DP: Collected_Information)

B. The following exception exit events are applicable to this PIC: CollectTimeout, CollectInfoFailure, InvalidInformation and O_Abandon.

- The CollectTimeout event is detected when enough information to process the call was not received by the SSF/CCF before a normal interdigit timer expires. For an SS7 trunk, this event corresponds to the IAM not containing the information necessary to process the call. In this case, there may be no timing involved (timing may be involved for ISUP overlap sending).
- The CollectInfoFailure event is detected when the SSF/CCF is unable to perform the information collection due to a lack of switch resources (e.g., no digit receivers are available).
- The InvalidInformation event occurs when the information received from the caller is not valid, for instance the information received violates the dialing plan in force.
- O_Abandon event, as described in the O_Null PIC.

Comment: Some digit analysis is required to determine the end of dialing. However, this analysis may be modeled as separable from the rest of digit analysis, which occurs in the Analyze_Information PIC. There is no intention to specify an implementation. However, a switch should externally present the separable view described for closed numbering plans. (See Note 1.)

In the case of ISDN en bloc sending, the receipt of a SETUP message causes the BCSM to pass through the Collect_Information PIC to the Collected_Information DP, without further processing in the

Collect_Information PIC. Note that the BCSM transitions to Collected_Information DP when the initial information package/dialing string is received from the calling party -- this occurs when enough information is received to proceed with call processing (e.g., after the NPA-NXX has been received via ISDN overlap sending when EAMF overlap outpulsing applies). Specifically, for the digit-by-digit collection case, if the Collected_Information DP is armed as a Trigger Detection Point-Request (TDP-R), the SSF sends Collected_Information operation to the SCF when enough information is received to determine if the TDP criteria are met. The SSF suspends BCSM processing but will collect further digits. It is network operator-specific to determine when complete information is available. (See Note 2.)

NOTE

1) This separable view is provided by supporting distinct DPs. The Collected_Information DP is used after digit collection and the Analyzed_Information DP is used after the rest of the digit analysis.

2) In some networks, it may be not possible for the CCF/SSF to determine when the called number information is complete. Therefore, TDP criteria for the Collected_Information DP may be met in such networks before the called number information is complete.

Corresponding T1.607 call state:

1. Call Initiated and (optionally)
2. Overlap Sending

5.3.2.1.4 Analyze_Information

Entry events:

Availability of complete initial information package/dialing string from originating party (DP: Collected_Information) or route busy event reported from the Select_Route PIC.

Function:

Information being analyzed and/or translated according to dialing plan to determine routing address and call type (e.g., local exchange call, transit exchange call, international exchange call).

One of the results of processing in this PIC is determination of routing address:

- *Called Party Number* only (*Called Party Number* is served by the SSF);
- *Called Party Number* and route index, where the route index is a pointer to a trunk group to route an out going call attempt on (*Called Party Number* is served by another SSF);
- *Called Party Number* and route index, where the route index is a pointer to a list of trunk groups to route an outgoing call attempt on (*Called Party Number* is served by another SSF).

Information Available:

After the SSF/CCF determines the information has been analyzed, the SSF/CCF has the following information available associated with the originating call portion:

- Charge Number, Calling Party Number, Additional Calling Party Number, Calling Party BGID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, Facility Information, and other feature-related information. This information is available for each access type under the conditions defined in the O_Null PIC.
- Analysis Results (of the Collected Information) - As described below.

From a non-ISDN line or DSS1 interface, this consists of one or more of the following:

- *Called Party Number* - As per dialing plan.
- Numbering Plan Indicator - see T1.113 Numbering Plan Indicator signaling information.
- Type Of Call - see clause 3.
- Carrier - see clause 3.
- Carrier Identification Code - see clause 3. Available for Inter-network carrier calls.
- Carrier Selection - see clause 3. Available for Inter Serving Area ID carrier calls.
- Collected Information - Access Code within a CDP, Feature Code, Feature Activation, Prefix, Carrier Access Code/Carrier Identification Code, Collected Address Information/Digits - as described under the Collect_Information PIC.

From a conventional or SS7 trunk interface, this consists of one or more of the following:

- Charge Number - as defined in the O_Null PIC (for an SS7 trunk) or Collected_Information PIC (for a conventional trunk when EAMF signaling is used).
- *Called Party Number* and Numbering Plan Indicator (as defined above for non-ISDN line or DSS1 interface).
- Carrier Identification - available for Inter Serving Area ID carrier calls.
- Carrier Selection - see clause 3. Available for Inter Serving Area ID carrier calls.
- Originating Line Information - see clause 3. Available for Inter Serving Area ID carrier calls.
- Route Index - see clause 3. Available if this call does not terminate on this SSF/CCF.
- Collected Information - Collected Address Information, Prefix, Carrier Identification Code, Feature Code, Facility Code - see description under the Collect_Information PIC.

From a private-facility trunk, this consists of one or more of the following, depending on the type of private-facility trunk:

- *Called Party Number* and Numbering Plan Indicator (as defined above)
- Type Of Call (as defined above)
- Carrier - see clause 3. Private network/facility, intra-serving area, or a specific Inter Serving Area ID or international).
- Carrier Identification Code - see clause 3. Available for internetwork carrier calls.
- Carrier Selection - see clause 3. Available for Inter Serving Area ID carrier calls.
- Traveling Class Mark - see clause 3. Available if received on the facility.
- Route List - see clause 3.
- Facility Restriction Level - as described under the Collect_Information PIC
- Collected Information - Collected Address Information/Digits, Access Code within a Customer Dialing Plan, Feature Code, Carrier Access Code, Prefix - see description under the Collect_Information PIC.

<u>Exit events:</u>

A. Availability of routing address and nature of address. (DP: Analyzed_Information)

B. The following exception exit events are applicable to this PIC: InvalidInformation and O_Abandon.

  - The InvalidInformation event (e.g., wrong number).

- The O_Abandon event, as described in the O_Null PIC.

Comments: Note that routing address does not necessarily mean that the final physical route has been determined (e.g., route list has not been searched, hunt groups have not yet been searched, directory number has not yet been translated to physical port address), though this may be the case (e.g., when routing to a specific private facility).

5.3.2.1.5 Select_Route

Entry Events

Availability of routing address and call type (DP: Analyzed_Information) or route failure event reported from the Send_Call or O_Alerting PIC.

Functions:

A. Routing address and call type are being interpreted. The next route is being selected. This may involve sequentially searching a route list, translating a directory number into physical port address, etc. The individual destination resource out of a resource group (e.g., a multi-line hunt group, a trunk group) is not selected. In some cases (e.g., an analog line interface), a single resource (not a group) is selected.

B. When the entry event is the route failure event from the Send_Call PIC (see below), the SSF/CCF must first check the Route Failure Condition 1, Route Failure Condition 2, or Route Failure Condition 3 as defined under the Send_Call PIC exit events. If these conditions are true, then the call shall proceed to the Analyze Information PIC in order to use the next *Called Party Number*.

C. If these three conditions are not met, then depending on the location in the network where the route is busy, the action is one of the following:

   - If the trunk group selected for the call is busy at this switch, the SSF/CCF attempts to route the call on the next trunk group that has been specified for the call (when a route list is being searched or alternate routes are specified by the SCF). Call processing moves to the Analyze_Information PIC when one of two conditions occurs: all private-facility trunk groups have been tried and routing over a public facility is allowed, or routing to a particular intra or internetwork carrier has been tried and an alternate carrier is allowed.
   - If all of the trunk groups (private and public) have been tried and no route is available, the Route_Select_Failure event is detected.
   - If route busy is detected at another switch, an indication of this condition may be received via SS7 signaling. In this case, a Route_Select_Failure event is detected.

Information Available:

After the SSF/CCF determines the route has been selected, the SSF/CCF has the following information available with restrictions as noted:

- Charge Number, Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, other feature-related information - This information is available for each access type under the conditions defined in the O_Null PIC.
- Analysis Results - See description in the Analyze_Information PIC.
- Routing Information - When more than one route has been specified for the call (either by the SCF or as part of the information stored at the SSF/CCF), the SSF/CCF remembers what routes

have been tried for this call and which route to select next. If the call is to an Inter Serving Area ID carrier, the routing information includes Circuit Code information.

Exit Events:

A. Route Selected event.

B. Unable to select a route (e.g., unable to determine a correct route, no more routes on route list) or indication from the terminating half BCSM that the call cannot be presented to the terminating party (e.g., network congestion) (DP: Route_Select_Failure).

C. Originating party abandons call. (O_Abandon event)

D. The route busy event leading to the Analyze_Information PIC as described for the functions of the Select_Route PIC.

5.3.2.1.6 Authorize_Call_Setup

Entry Events:

Route Selected event.

Function:

The authority of the calling party to place this particular call is verified.

Information Available:

After the SSF/CCF determines the call setup has been authorized, the SSF/CCF has the following information available with restrictions as noted:

A. Charge Number, Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, other feature-related information - This information is available for each access type under the conditions defined in the O_Null PIC.

B. Analysis Results - See description in the Analyze_Information PIC

C. Routing Information - See description in the Select_Route PIC

D. Facility Restriction Level - as described under the Collect_Information PIC

Exit Events:

A. Call Setup Authorized event. The Call Setup Authorized event occurs when the authority to place the call is verified. For an SS7-supported trunk interface, if the received IAM indicates that a continuity check is being performed on the call connection and the call terminates to a non-ISDN line or ISDN interface, the Call Setup Authorized event occurs when an ISUP Continuity Message (COT) with a successful indication is received.

B. Originating party abandons call (O_Abandon event).

C. The Authorization Route Failure event occurs when the authority to place the call is denied (e.g., business group restriction mismatch, toll restricted calling line). For an SS7-supported trunk interface, the Authorization Route Failure event occurs when a COT with a failure indication is received. This event causes a BCSM transition to O_Exception.

5.3.2.1.7 Send_Call

Entry Events:

Call Setup Authorized event.

Functions:

A. The SSF/CCF sends an indication of the desire to set up a call to the specified Called Party ID to the terminating call portion. The information that may be passed to the terminating call portion is: Charge Number, Calling Party ID, Calling Party BGID, Calling Party Category (determined by the Class of Service information or ISUP originating line information parameter), Bearer Capability, Called Party ID, Calling Party Subaddress, Called Party Subaddress, Carrier, Route Index, Carrier Identification Code, Circuit Code, Carrier Selection, and Traveling Class Mark (TCM). Other feature-information not used by the processing modeled by this PIC (e.g., call forwarding, generic name, and business group information) may also be passed to the terminating call portion.

B. For an ISDN caller, during the processing that this PIC models, the SSF/CCF returns a CALL PROCeeding message. For non-ISDN lines, conventional trunks, and private-facility trunks, no treatment is applied.

C. For SS7-supported trunks, if the received IAM indicates a continuity check is required in this circuit, the procedures for performing the continuity check are followed. If the continuity check is successful, an indication of continuity success is passed to the terminating call portion. If the received IAM indicates a continuity check is being performed on a previous circuit in the connection, upon receiving a COT with a successful indication, this indication is passed to the terminating call portion.

Information Available:

After the SSF/CCF determines the call has been delivered (to the terminating half), the SSF/CCF has the following information available with restrictions as noted:

A. Charge Number, Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, other feature-related information - This information is available for each access type under the conditions defined in the O_Null PIC.

B. Analysis Results - See description in the Analyze_Information PIC

C. Routing Information - See description in the Select_Route PIC

D. Facility Restriction Level - as described under the Collect_Information PIC

Exit events:

A. A route failure event is detected when :

- an indication of a T_Busy event specifying route busy; or
- a Call Rejected event specifying route busy (received when the route is found to be busy at a switch other than the local switch) is received from the terminating call portion. In both cases, the originating call portion returns to the Select_Route PIC. This event is not detected at a DP in T1 IN.
- the following two conditions are met, hereafter called Route Failure Condition 1:

A. an indication of a T_Busy event specifying route busy (received when the route at the local switch is found to be busy) is received from the terminating call portion (presentation failure event from the Present_Call PIC);

B. The route was determined by switch translations at the Analyze_Information PIC.

- the following two conditions are met, hereafter called Route Failure Condition 2:

i) a Call Rejected event specifying route busy (received when the route is found to be busy at a switch other than the local switch) is received from the terminating call portion (presentation failure event from the Present_Call PIC)

The SSF/CCF shall interpret the cause values from the signaling network to determine the route busy condition. The following cause values received in an SS7 message should be interpreted as route busy.

3	No route to destination
25	(ASNI) Exchange routing error
29	Facility rejected
34	No circuit/channel available
38	Network out of order
41	Temporary failure
42	Switching equipment congestion
47	Resource unavailable, unspecified
95	Invalid message, unspecified
102	Recovery on timer expiry
111	Protocol error, unspecified
127	Interworking, unspecified.

ii) The route was determined by the switch translations at the Analyze Information PIC;

- the following condition is met, hereafter called Route Failure Condition 3:

A O_Called_Party_Busy event or O_No_Answer event occurs (as specified below);

NOTE - The Route_Failure event takes precedence over the O_Called_Party_Busy and O_No_Answer events.

In all five cases, the originating call portion returns to the Select_Route PIC. This event is not detected at a DP in T1 IN.

B. An O_Answer event occurs when an indication of a T_Answer event is received from the terminating call portion. This event causes call processing to move to the O_Answer DP.

C. An O_Term_Seized event occurs when an indication of a call accepted event is received from the terminating call portion or when certain abnormal cases occur in ISDN when the call is offered to an ISDN interface and no user equipment has responded, but an “indication to apply audible ringing” is sent from the terminating call portion to the originating call portion and as a result, audible ringing is to be sent to the caller. In ISDN this occurs, for example, when a call is offered to the interface with a SETUP message and the call setup timer T303 is initiated. If T303 expires and no response is received, the switch retransmits the SETUP message, reinitializes T303, and sends inband audible ringing back to the caller. If the call is from an ISDN user, a PROGress message is also sent containing progress indicator #10, “delay in response at called interface” and progress descriptor #8, “inband tone or pattern not available”. In this case, the calling party may receive inband audible ringing, however, the called party has not accepted the call. When the O_Term_Seized event occurs, the treatment applied depends on the originating access type.

No additional actions are taken in a non-ISDN line or private facility trunk.

For a call originating from an ISDN interface, the caller also receives an ALERTing message or under certain conditions a PROGress message containing progress indicator information set to “inband information or pattern now available”

For SS7-supported trunks, an Address Complete Message (ACM) is sent.

In these cases, audible ringing, if applicable, is being sent from the originating call portion of the terminating switch.

D. If the O_No_Answer timer expires or an indication of the T_No_Answer event is received before an O_Answer event is detected (i.e., before the called party answers), the SSF/CCF reports the event to the SCF.

E. The O_Called_Party_Busy event occurs when an indication of a T_Busy event specifying user busy is received from the terminating portion of the call (i.e., network-determined-user-busy). This event also occurs when an indication of a Call Rejected event specifying user busy (i.e., user-determined-user busy) is received from the terminating portion of the call. For calls originating from non-ISDN lines, conventional trunks, and private-facility trunks, if an indication of busy is received from the terminating portion of the call and no originating triggers or requested events apply, busy tone is provided.

In addition to these busy events, the following “Call Rejected” conditions are also treated as O_Called_Party_Busy events:

- a Termination Denied event is received from the terminating call portion (Authorize_Termination_Attempt PIC); or,
- an indication of a Call Rejected event is received from the terminating call portion (T_Alerting PIC) that does not specify busy. An example of a Call Rejected event that does not specify busy is the case when a call is offered to an ISDN interface and either

  (a) no user has responded with an ALERTing or CONNect message when the call setup timers expire, or

  (b) the terminating user returns a call clearing message with a cause value of something other than “busy”.

  In this case, the terminating portion of the call is cleared.

F. For SS7-supported trunk interface, the authorization_route_failure event occurs when the continuity check procedure results in failure. This event causes a BCSM transition to the O_Exception.

G. Originating party abandons call (O_Abandon event).

H. A service feature request is received from the originating party: e.g., hook-flash, ISDN feature activator, or an O_DTMF_Entered event occurs (the SSF/CCF shall continue the normal call establishment procedure and send an O_DTMF_Entered message to the SCF). (DP: O_Mid_Call).

5.3.2.1.8 O_Alerting

Entry Event:

O_Term_Seized event (DP: O_Term_Seized DP)

Function:

A. Wait for the terminating party to answer. At this point, the caller receives inband audible ringing (from the terminating switch). Inband audible ringing is only provided for a subset of ISDN circuit mode calls. For a call originating from an ISDN interface, the caller also receives an ALERTing message or, under certain conditions, a PROGress message containing progress indicator information set to “inband information or pattern now available”.

B. An indication of a call progress event may be received from the terminating call portion. This may result in a Call Progress (CPG) message being sent on an SS7-supported trunk (if the originating access is an SS7-supported trunk) or an ALERTing or PROGress message being sent on an ISDN interface (if the originating access is an ISDN interface)

Information Available:

When the SSF/CCF is in this PIC, the SSF/CCF has the following information available with restrictions as noted:

A. Charge Number, Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, other feature-related information - This information is available for each access type under the conditions defined in the O_Null PIC.

B. Analysis Results - See description in the Analyze_Information PIC.

C. Routing Information - See description in the Select_Route PIC.

D. Facility Restriction Level - as described under the Collect_Information PIC.

Exit Events:

A. The O_Answer event occurs when an indication of a T_Answer event is received from the terminating portion of the call (e.g., terminating party goes off hook, T1.607 CONNect message received, ISUP Answer message received) (DP: O_Answer). When the O_Answer event occurs, the treatment applied is described in the Send_Call PIC.

B. A service feature request is received from the originating party: e.g., hook-flash, ISDN feature activator, or An O_DTMF_Entered event occurs (the SSF/CCF shall continue the normal call establishment procedure and send an O_DTMF_Entered message to the SCF). (DP: O_Mid_Call).

C. A route failure event is detected when the following condition is met, hereafter called Route Failure Condition 3:

- O_Called_Party_Busy event or O_No_Answer event occurs (as specified below).

NOTE - The Route Failure event takes precedence over the O_Called_Party_Busy and O_No_Answer events.

In this case, the originating call portion returns to the Select_Route PIC. This event is not detected at a DP in T1 IN.

D. The O_No_Answer event from this PIC is the same as the O_No_Answer event defined as an Exit Event from the Send_Call PIC. (DP: O_No_Answer).

E. From this PIC, the O_Called_Party_Busy event occurs either when:

- a Call Rejected event specifying user busy is received, or
- when an indication of a Call Rejected event not specifying busy is received from the terminating call portion (as described in the Send_Call PIC).

In addition, for a call to an ISDN user, after the SETUP message is offered and an ALERTing message has been received (i.e., the terminating call portion is in the T_Alerting PIC), the ISDN user may reject the call. This Call Rejected event is treated as an O_Called_Party_Busy event by the originating call portion (DP: O_Called_Party_Busy).

F. Originating party abandons call (O_Abandon event).

5.3.2.1.9 O_Active

Entry Event:

Indication from the terminating half BCSM that the call is accepted and answered by terminating party. (DP: O_Answer)

Function:

In this PIC several processes may be initiated:

- Connection established between originating and terminating party. Message accounting/charging data may be being collected. Call supervision is being provided.
- The called party may be put on hold and returned to the active phase by service logic.
- The called party may be put on hold by a service logic and when the calling party disconnects, the calling party can be reconnected to the held call. The calling user receives appropriate information (e.g., re-ring) and a reconnection timer is applied.

Information Available:

Once the SSF/CCF has received an indication from the terminating half BCSM that the call has been answered, the SSF/CCF has the following information available with restrictions as noted:

- Information as per the O_Alerting PIC
- Feature Activation - A service or feature request from the originating party (e.g., DTMF, hook flash, ISDN feature activator, T1.607 HOLD or RETrieve message).

Exit events:

A. A service/service feature request is received from the originating party (e.g., DTMF, hook flash, ISDN feature activator, T1.607 HOLD or RETrieve message), A Timeout event occurs (the SSF/CCF shall continue the stable call and send a Timeout EDP-R message to the SCF), An O_DTMF_Entered event occurs (the SSF/CCF shall continue the stable call and send an O_DTMF_Entered message to the SCF), or a new reply from calling party (Reconnect procedure). (DP: O_Mid_Call)

B. A disconnect indication is received from the terminating party via the terminating half BCSM. (DP: O_Suspend). A disconnect timing is associated with this BCSM transition.

C. A disconnect indication (e.g., onhook, T1.607 disconnect message, SS7 release message) is received from the originating party. (DP: O_Disconnect)

E. An indication of expiration of the reconnection timer is received. (O_Exception).

F. A connection failure occurs (O_Exception)

Comments: Disconnect treatment and timing is different for call attempts originating from ISDN and analog line interfaces.

Corresponding T1.607 call state:

10. Active

T1.607 call states corresponding to disconnect:

11. Disconnect request,

12. Disconnect indication and

19. Release request.

5.3.2.1.10 O_Suspended

Entry event:

A suspend indication is received from the T_BCSM when the terminating party has disconnected (e.g., on hook). (DP: O_Suspended).

Function:

A. The connection between the originating and terminating party is maintained and depending on the incoming network connection, appropriate backward signaling takes place.

- In case that a disconnect indication is received from the T_BCSM, this PIC is immediately exited to the O_Disconnect DP without any action. As an option, the call can be continued for an appropriate period in order to offer follow-on initiated by O_Mid_Call.
- If the re-answer indication from the T_BCSM is received, the originating and terminating parties are reconnected.

B. The called party may be put on hold and returned to the active phase by a service logic.

C. The called party may be put on hold by a service logic and when the calling party disconnects, the calling party can be reconnected to the suspended call. The calling user receives appropriate information (e.g., re-ring) and a reconnection timer is applied. The timer may have been started in the active phase.

Information Available:

The information available in this PIC is the same as the information available in the O_Active PIC.

Exit events:

- Connection to the terminating party is resumed. The O_BCSM returns to the O_Active PIC.
- A service feature request is received from the originating party: e.g., hook flash, ISDN feature activator of facility or a new reply from calling party. (DP: O_Mid_Call).
- A disconnection indication is received from the originating party. (DP: O_Disconnect).
- A disconnection indication is received from the terminating party. (DP: O_Disconnect).
- An indication of expiration of the timer waiting for re-answer request is received from the T_BCSM. (DP: O_Disconnect)
- A trigger at O_Mid_Call is not initiated during an appropriate period. (DP: O_Disconnect).
- An indication of expiration of the calling party reconnection timer is received. (O_Exception).
- An exception event is encountered (O_Exception).

NOTES

- A Call Retention timer may exist. Disconnect treatment and timing are different for call reconnection, call suspension and call retention.
- After the release of the outgoing connection, the originating party may initiate another call (e.g., follow-on calling).

5.3.2.1.11 O_Exception

Entry event:

An exception condition is encountered (as described above for each PIC)

Function:

Default handling of the exception condition is being provided. This includes general actions necessary to ensure no resources remain inappropriately allocated, such as:

- If any relationships exist between the SSF and SCF(s), send an error to the SCF(s) closing the relationships and indicating that any outstanding call handling instructions will not run to completion.

  NOTE - This should be handled in the physical plane via an ABORT protocol to close the relationship (i.e., close the TCAP transaction) and indicate that any outstanding operations will not be run to completion.
- If an SCF previously requested that call parameters be provided at the end of the call (see the call information request operation in clause 7), these should be included in the error information.
- The SSF/CCF should make use of vendor-specific procedures to ensure release of resources within the SSF/CCF so that line, trunk, and other resources are made available for new calls.

Information Available:

Once the SSF/CCF has determined an exception condition has occurred, the SSF/CCF has information available as when the exception within the PIC occurred.

Exit event:

Default handling of the exception condition by the SSF/CCF completed (BCSM transition to O_Null PIC).

5.3.2.2 Terminating BCSM for T1 IN

The terminating half of the BCSM corresponds to that portion of the BCSM associated with the terminating party (see Figure 4).

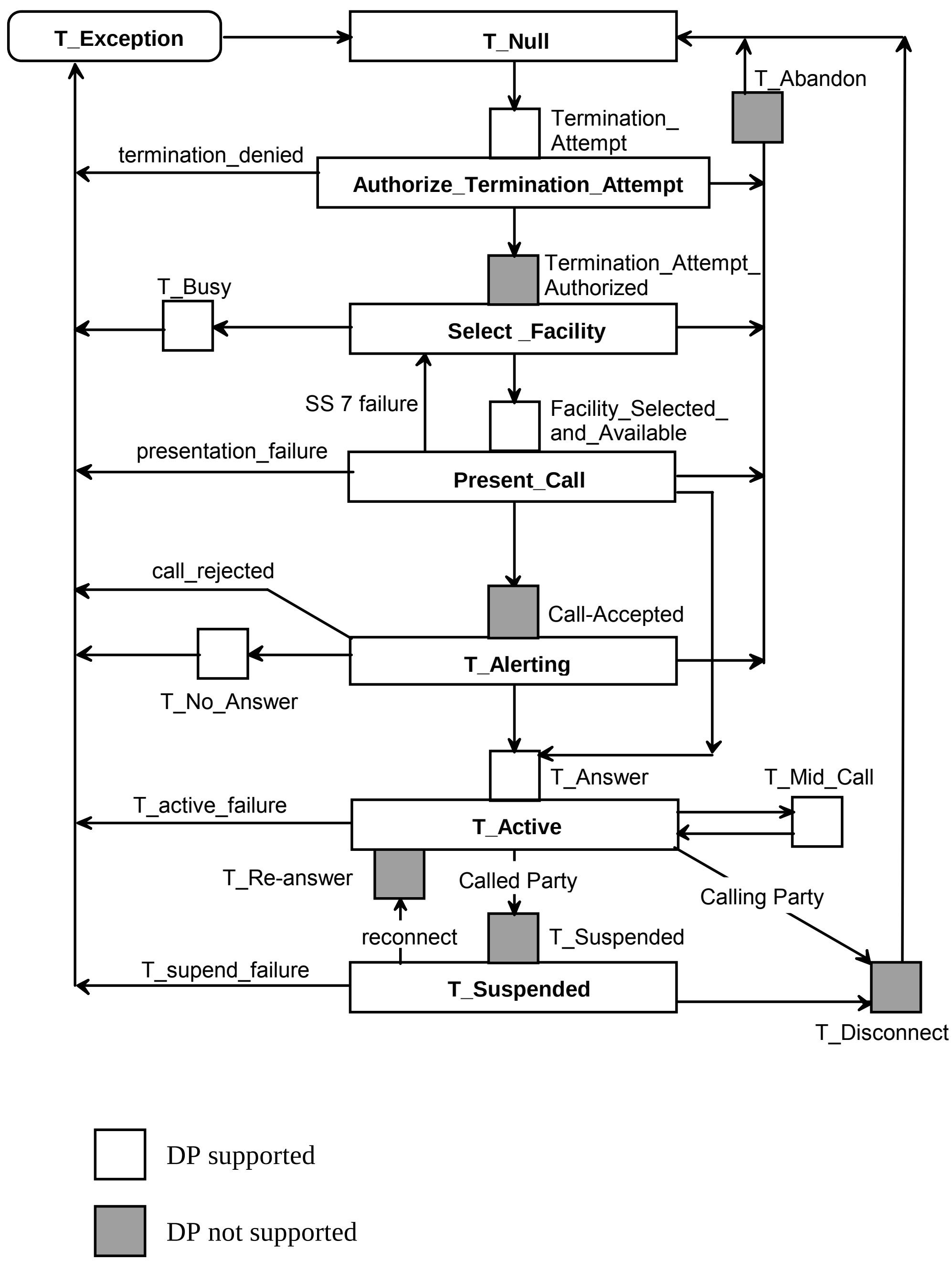


Figure 4 - Terminating BCSM for T1 IN

The following information is available at all PICs in the T-BCSM:

- Service Address Information - see clause 3
- Call Gapping Encountered - indicates the code/call control in effect at the SSF.

- Location Number - see ITU-T Recommendation Q.763. Used if the calling party is a mobile subscriber.
- Original *Called Party Number* - see T1.113. This information is received from the originating call portion.
- Redirecting Party ID - see T1.113. This information is received from the originating call portion.
- Redirection Information - see T1.113. This information is received from the originating call portion.
- ISDN Access Related Information: - See T1.113 Access Transport Parameter.

The description of each of the PICs in the terminating half of the BCSM follows.

5.3.2.2.1 T_Null

Entry event:

Disconnect and clearing of a previous call (T_Disconnect or T_Abandon), or default handling of exceptions by SSF/CCF completed.

Function:

Interface (line/trunk) is idled (no call exists, no call reference exists, etc.). Supervision is being provided.

Information Available:

Once the SSF/CCF has detected the Termination_Attempt event, the following information is available and associated with the terminating portion of the call with restrictions as noted (information associated with the originating portion of the call as per the Send_Call PIC is still available):

a) Charge Number, Calling Party Number; Calling Party Business Group ID, Bearer Capability - Available for each access type under conditions identified in the O_Null PIC. This information is received from the originating call portion.

b) Calling Partys Category - see T1.113. Determined by the Class of Service information, ISDN User Part originating line information parameter (see clause 3), or information from EAMF signaling.

c) *Called Party Number*, Carrier, Carrier Identification Code, Circuit Code, Carrier Selection; Route Index; and Traveling Class Mark - This information is received from the originating call portion.

d) Class of Service of Terminating Access - see clause 3. This is either a Customer Class of Service, a Trunk Class of Service, or a Private-Facility Class of Service for the Terminating Access (Dialed Number, Circuit, or trunk group).

e) Called Party Sub-Address - See T1.607.

f) Calling Party Sub-Address - See T1.607.

g) Called Party BGID - see clause 3. This information is determined in this PIC when the terminating party is a member of a Business Group and is served by a non-ISDN line or DSS1 interface on this SSF/CCF.

Exit event:

- Indication of incoming call received from originating half BCSM. (DP: Termination_Attempt)

- The following exception exit event is applicable to this PIC: T_Abandon. If the call encounters T_Abandon during PIC processing, the exception event is not visible because there is no corresponding DP.
- The T_Abandon occurs when an indication of call disconnection is received from the originating part of the call.

Corresponding T1.607 call state:

0. Null

5.3.2.2.2 Authorize_Termination_Attempt

Entry Event:

Termination_Attempt event. (DP: Termination_Attempt).

Function:

Verifies the authority to route this call to the terminating access (e.g., DN or trunk group), e.g., check business group restrictions, restricted incoming access to line, or bearer capability compatibility.

Information Available:

The SSF has the same information available for the terminating call portion after the Termination_Attempt_Authorized event is detected as it does when the Termination_Attempt event is detected.

Exit Events:

- Termination_Attempt_Authorized event. This event occurs when the switch has verified the authority to terminate the call to the terminating access (Termination_Attempt_Authorized).
- The Termination Denied event occurs when the authority to route the call to the terminating user is denied. (This causes a BCSM transition to T_Exception.)
- The T_Abandon event occurs when an indication of clearing is received from the originating portion of the call (T_Abandon).

5.3.2.2.3 Select_Facility

Entry Event:

Termination_Attempt_Authorized event or an SS7 failure occurs causing a reattempt. The SS7 failure in the Present_Call can be caused by a timer expiry upon sending the first Circuit Reservation Message (CRM) or a continuity check failure.

Function:

The busy/idle status of the terminating access is determined.

- For a non-ISDN line, if the line is already involved with an existing call, the line is treated as network-determined user busy.

- For a call terminating to an ISDN interface (on a non-shared DN/CT), network-determined user busy is the detection of one or more of the following conditions:
- Interface busy: That is, a B-channel is not available for the call.
- Call-reference busy: There are no idle call reference values available on the terminating DN/CT with which the call can be offered.
- In addition, if the terminating DN is associated with an Multi-Line Hunt Group (MLHG), busy means that no hunt-terminals within the group are available and the queue, if any, is full.
- For conventional trunks, SS7-supported trunks, and private-facility trunks, busy is when all trunks within the selected trunk group are busy.

Information Available:

When the Facility_Selected_and_Available event is detected, the following information is available and associated with the terminating portion of the call with restrictions as detailed noted:

a) Information as per the T_Null PIC.

b) Facility Group - see clause 3. For calls routed out of this SSF/CCF, this identifies the Trunk Group (private or public) that has been selected on which to route the call. For calls terminating to a non-ISDN line or DSS1 interface within the SSF/CCF, this may identify a particular Multi-line Hunt Group.

c) Facility Group Member - see clause 3. For calls out of this SSF/CCF, this identifies the trunk (private or public) that has been selected on which to route the call. For calls terminating to a non-ISDN line DSS1 interface on the SSF/CCF, this may identify the hunt-terminal within the Multi-line Hunt Group that has been selected for this call.

Exit events:

- Facility_Selected_and_Available event: This event occurs when the terminating access is not busy (i.e., an idle facility [e.g., B-channel, call appearance, or trunk] could be found). (DP: Facility_Selected_and_Available).
- A T_Busy event occurs when the terminating access is busy (as defined above). The T_Busy event may also be detected as a result of an analog line being out of order, marked as busy by a customer make-busy key, or as a result of certain maintenance actions. (DP: T_Busy)
- After detecting T_Busy, if IN service logic is not needed on the call and no switch-based features apply, an indication of the T_Busy event describing the type of busy (e.g., user or network) is passed to the originating call portion. If a terminating feature acts on the T_Busy event and changes the event (e.g., as in the Call Waiting feature), the event is not passed to the Originating BCSM.
- The T_Abandon event occurs when an indication of clearing is received from the originating portion of the call

5.3.2.2.4 Present_Call

Entry event:

Facility_Selected_and_Available event. (DP: Facility_Selected_and_Available)

Functions:

Terminating resource informed of incoming call (e.g., line seizure, SETUP message, ISUP IAM message). In the case of an analog line, ringing is applied.

Information Available:

When the Call Accepted event is detected, the following information is available and associated with the terminating portion of the call with restrictions as noted:

- Information as per the T_Null PIC.
- Facility Group, Facility Group Member - See description in the Select Facility PIC.
- Information regarding the call connection - This information includes whether the call is end-to-end SS7 or not and whether the originating access is ISDN or non-ISDN.

Exit events:

- Terminating party is being alerted (e.g., ringing being applied, T1.607 ALERTing message, ISUP ACM message) (Call_Accepted).
- Call is accepted and answered by terminating party (e.g., terminating party goes off hook, T1.607 CONNect message received, ISUP answer message received) (DP: T_Answer)
- Indication of originating party abandon received from originating half BCSM. (T_Abandon)
- The T_Abandon as described in T_Null PIC.
- A timer expiry upon sending the first Circuit Reservation Message (CRM) or a continuity check failure. (SS7 failure). This event causes call processing to move to the Select Facility PIC.
- Presentation Failure: The call cannot be presented, (e.g., ISDN user determined busy, ISUP release message with busy cause). This event causes call processing to move to the T_Exception and is notified to the originating call portion (Send Call PIC).

Corresponding T1.607 call state:

6. Call present

5.3.2.2.5 T_Alerting

Entry event:

Terminating party is being alerted of incoming call (Call Accepted).

Function:

An indication is sent to the originating half BCSM that the terminating party is being alerted. Continued processing of call setup (e.g., ringing, audible ring indication) is taking place while waiting for the call to be answered by terminating party.

Information Available:

Once the terminating party is being alerted of the incoming call, the information as per Present_Call PIC is available.

Exit events:

- Call is accepted and answered by terminating party (e.g., terminating party goes off hook, T1.607 CONNect message received, ISUP answer message received). (DP: T_Answer)
- The following exception exit events are applicable to this PIC: call rejected, T_No_Answer and T_Abandon.
    - a) Call rejected exception event may happen when an ISDN user rejects a call while being alerted.
    - b) The T_No_Answer event occurs when the terminating party does not answer before the switch-based ringing timer expires. An indication of T_No_Answer event is passed to the originating half of the BCSM. This event leads to the T_No_Answer DP.
    - c) Indication of originating party abandon received from originating half BCSM. (T_Abandon).
    - d) The T_Abandon as described in the T_Null PIC.

Comment: For terminations to SS7 trunk groups, this PIC is entered upon the receipt of an address complete (ACM) message.

Corresponding T1.607 call states:

7. Call received

8. Connect request

5.3.2.2.6 T_Active

Entry Events:

Call is accepted and answered by terminating party (e.g., terminating party goes off hook, T1.607 Connect message received, ISUP answer message received) (DP: T_Answer).

Function:

In this PIC several processes may be initiated:

- An indication is sent to the origination half BCSM that the terminating party has accepted and answered the call. Connection established between originating and terminating party. Call supervision is being provided.
- The calling party may be put on hold and returned to the active phase by a service logic.
- The calling party may be put on hold by a service logic and when the called party disconnects, the called party can be reconnected to the held call. The called user receives appropriate information (e.g., re-ring) and a reconnection timer is applied. The timer may have been started in the active phase.

Information Available:

Once the call is accepted and answered by the terminating party, the following information is available and associated with the terminating portion of the call with restrictions as noted:

- a) Information as per T_Alerting

b) Feature Activation - see T1.610. For this PIC, a service or feature request from the terminating party (e.g., DTMF, hook flash, ISDN feature activator, T1.610 HOLD or RETRIEVE message).

Exit events:

- A service/service feature request is received from the terminating party (e.g., DTMF, hook flash, ISDN feature activator, T1.607 HOLD or RETrieve message) (DP: T_Mid_Call).
- A disconnect indication (e.g., onhook, T1.607 disconnect message, SS7 release message) is received from the terminating party (T_Suspend).
- A disconnect indication is received from the originating party via the originating half BCSM (T_Disconnect).
- An indication of expiration of the reconnection timer is received (T_Exception).
- A connection failure occurs (T_Exception).

Comments

- Disconnect indications and treatment are asymmetrical in the way disconnect timing is applied.

Corresponding T1.607 call state:

10. Active

T1.607 call states corresponding to T_Disconnect:

11. Disconnect request

12. disconnect indication

19. Release request

5.3.2.2.7 T_Suspended

Entry event:

An indication is received from the outgoing network that the terminating party has disconnected (e.g., on hook). (T_Suspend)

Function:

The physical resources associated with the call remain connected.

According to the received indication the following applies:

- A suspend indication is sent to the originating half BCSM.
- For an SS7-supported trunk or an ISDN interface, in case that a disconnect indication (e.g., T1.607 disconnect message, SS7 release message) is received from the terminating party, this PIC is immediately exited to the T_Null PIC.
- In the following cases, the timer is started and the call waits for re-answer request from the terminating party.

a) For an SS7 supported trunk, in case of receiving network initiated suspend message.

b) For an analog interface, in case of detecting on-hook.

- If re-answer request (e.g., off-hook, SS7 resume message) is received from the terminating party before the timer expires, the originating and terminating parties are reconnected.

Note - Both a Call Resume timer and a Call Retention timer may exist in this PIC. IN implementations may use a single timer for both conditions.

Information Available:

While in the T_Suspended PIC, the SSF has the same information available for the terminating call portion as it has in the T_Active PIC.

Exit event:

- The terminating party re-answers or a resume message is received before the timer expires; the T_BCSM returns to the T_Active PIC. (T_Re-answer).
- The timer expires (T_Disconnect).
- A disconnection indication is received from the terminating party (T_Disconnect).
- A disconnection indication is received from the originating party (T_Disconnect).
- An exception event is encountered (T_Exception).

5.3.2.2.8 T_Exception

Entry event:

An exception condition is encountered (as described above for each PIC).

Function:

An indication of the exception condition is sent to the originating half BCSM. Default handling of the exception condition is being provided. This includes general actions necessary to ensure no resources remain inappropriately allocated, such as:

- If any relationships exist between the SSF and SCF(s), send an error to the SCF(s) closing the relationships and indicating that any outstanding call handling instructions will not be run to completion (see Note 1).
- If an SCF previously requested that call parameters be provided at the end of the call (see the call information request operation), these should be included in the error.
- The SSF/CCF should make use of vendor-specific procedures to ensure release of resources within the SSF/CCF so that line, trunk, and other resources are made available for new calls.

NOTE 1 - This should be handled in the physical plane via an ABORT protocol procedure to close the relationship (i.e., close the TCAP transaction) and indicate that any outstanding operations will not be run to completion)

Information Available:

Once the SSF/CCF has determined an exception condition has occurred, the SSF/CCF has information available as when the exception within the PIC occurred.

Exit event:

Default handling of the exception condition by SSF/CCF completed (BCSM transition to T_Null PIC).

5.3.2.3 BCSM transitions in the T1 IN call model

Table 1 and Table 2 describe the complete set of allowable BCSM transitions for the T1 IN originating and terminating call models. Not all allowable transitions are depicted in the BCSM diagrams.

Table 1 - Complete Set of Originating BCSM transitions

Complete Set of Originating BCSM transitions	
From	**To**
Origination_Attempt DP	Authorize_Origination_Attempt PIC Collect_Information PIC Analyze_Information PIC Select_Route PIC
Collected_Information DP	Collect_Information PIC Analyze_Information PIC Select_Route PIC
Analyzed_Information DP	Collect_Information PIC Analyze_Information PIC Select_Route PIC
O_Term_Seized DP	O_Alerting
Route_Select_Failure DP	Collect_Information PIC Analyze_Information PIC Select_Route PIC O_Exception
O_Called_Party_Busy DP	Collect_Information PIC Analyze_Information PIC Select_Route PIC O_Exception
O_No_Answer DP	Collect_Information PIC Analyze_Information PIC Select_Route PIC O_Exception
O_Answer DP	O_Active PIC
O_Suspended DP	O_Suspended PIC
O_Mid_Call DP	Send_Call PIC
O_Mid_Call DP	O_Alerting PIC
O_Mid_Call DP	O_Active PIC Analyze_Information PIC Select_Route PIC
O_Mid_Call DP	O_Suspended PIC Analyze_Information PIC Select_Route PIC

Complete Set of Originating BCSM transitions	
From	**To**
O_Disconnect DP	O_Null PIC Collect_Information PIC Analyze_Information PIC Select_Route PIC
O_Null PIC	Origination_Attempt DP
Authorize_Origination_Attempt PIC	Collect_Information PIC O_Exception O_Null PIC
Collect_Information PIC	Collected_Information DP O_Null PIC O_Exception
Analyze_Information PIC	Analyzed_Information DP O_Null O_Exception
Select_Route PIC	Analyze_Information PIC Authorize_Call_Setup PIC Route_Select_Failure DP O_Null PIC
Authorize_Call_Setup PIC	Send_Call PIC O_Null PIC O_Exception
Send_Call PIC	O_Term_Seized DP O_Mid_Call DP O_Called_Party_Busy DP O_Answer DP O_No_Answer DP Select_Route PIC O_Null PIC O_Exception
O_Alerting PIC	Select_Route PIC O_Mid_Call DP O_Answer DP O_No_Answer DP O_Called_Party_Busy DP O_Null PIC
O_Active PIC	O_Midcall DP O_Disconnect DP O_Suspended DP O_Exception
O_Suspended PIC	O_Active PIC O_Mid_Call DP O_Disconnect DP O_Exception
O_Exception	O_Null PIC

Table 2 - Complete Set of Terminating BCSM transitions

Complete Set of Terminating BCSM transitions	
From	**To**
Termination_Attempt DP	Authorize_Termination_Attempt PIC Select_Facility PIC
Facility_Selected_and_Available DP	Present_Call PIC
T_Busy DP	Select_Facility PIC Present_Call PIC T_Exception
T_No_Answer DP	Select_Facility PIC T_Exception
T_Answer DP	T_Active PIC
T_Midcall DP	T_Active PIC
T_Null PIC	Termination_Attempt DP
Authorize_Termination_Attempt PIC	Select Facility PIC T_Null PIC T_Exception
Select_Facility PIC	Facility_Selected_and_Available DP T_Busy DP T_Null PIC
Present_Call PIC	T_Answer DP T_Alerting PIC Select_Facility PIC T_Null PIC T_Exception
T_Alerting PIC	T_Answer DP T_No_Answer DP T_Null PIC T_Exception
T_Active PIC	T_Midcall DP T_Null PIC T_Suspended PIC T_Exception
T_Suspended PIC	T_Active PIC T_Null PIC T_Exception
T_Exception	T_Null PIC

5.3.3 BCSM indications for the T1 IN call model

5.3.3.1 User - O_BCSM access signaling indications (Category 1)

Definition: These Indications include the representation of the network's perception of possible actions taken by the calling party as well as the calling party's perception of actions taken by the network. The Indications are between a user (i.e., calling party) and a local exchange that is originating a call. They include the definition of how actions by the user (originating call model) affect the originating call model (user). These Indications are derived from Access Signaling (e.g., DSS1, analog) as well as any other information that is available. Figure 5 illustrates these indications.

Indications:

(1) An Indication is sent from User to O_BCSM to initiate call establishment (e.g., SETUP).

(2) An Indication is sent from O_BCSM to User that network is unable to initiate call (e.g., RELEASE COMPLETE).

(3) An Indication is sent from O_BCSM to User acknowledging the call initiation Indication (e.g., SETUP ACKNOWLEDGE).

(4) The User sends call (dialing) information to the O_BCSM (e.g., INFORMATION).

(5) An Indication is sent from O_BCSM to the User to terminate the sending of call information (e.g., CALL PROCEEDING).

(6) An Indication is sent from the User to the O_BCSM upon completion of call information.

(7) User is informed that call has been routed to another environment or network (e.g., PROGRESS).

(8) An Indication is sent from the O_BCSM to the User when the called party is being alerted (e.g., ALERTING, CONNECT).

(9) An Indication is sent from the O_BCSM to the User when the call is accepted.

(10) The User acknowledges that the call is accepted.

(11) The O_BCSM sends an Indication to the User that the called party is unable to accept the call, due to busy condition.

(12) The O_BCSM sends an Indication to the User since the called party is unable to accept the call, due to no answer condition.

(13) An Indication is received by the O_BCSM from the User to end the call.

(14) The O_BCSM indicates to the User that the call is being disconnected.

(15) The User acknowledges to the O_BCSM that the call is being disconnected.

(16) An Indication is sent to the user when the connection towards the Called Party is suspended.

(17) An Indication is sent to the user when the connection towards the Called Party is reconnected.

(18) The user sends DTMF digits to the O_BCSM.

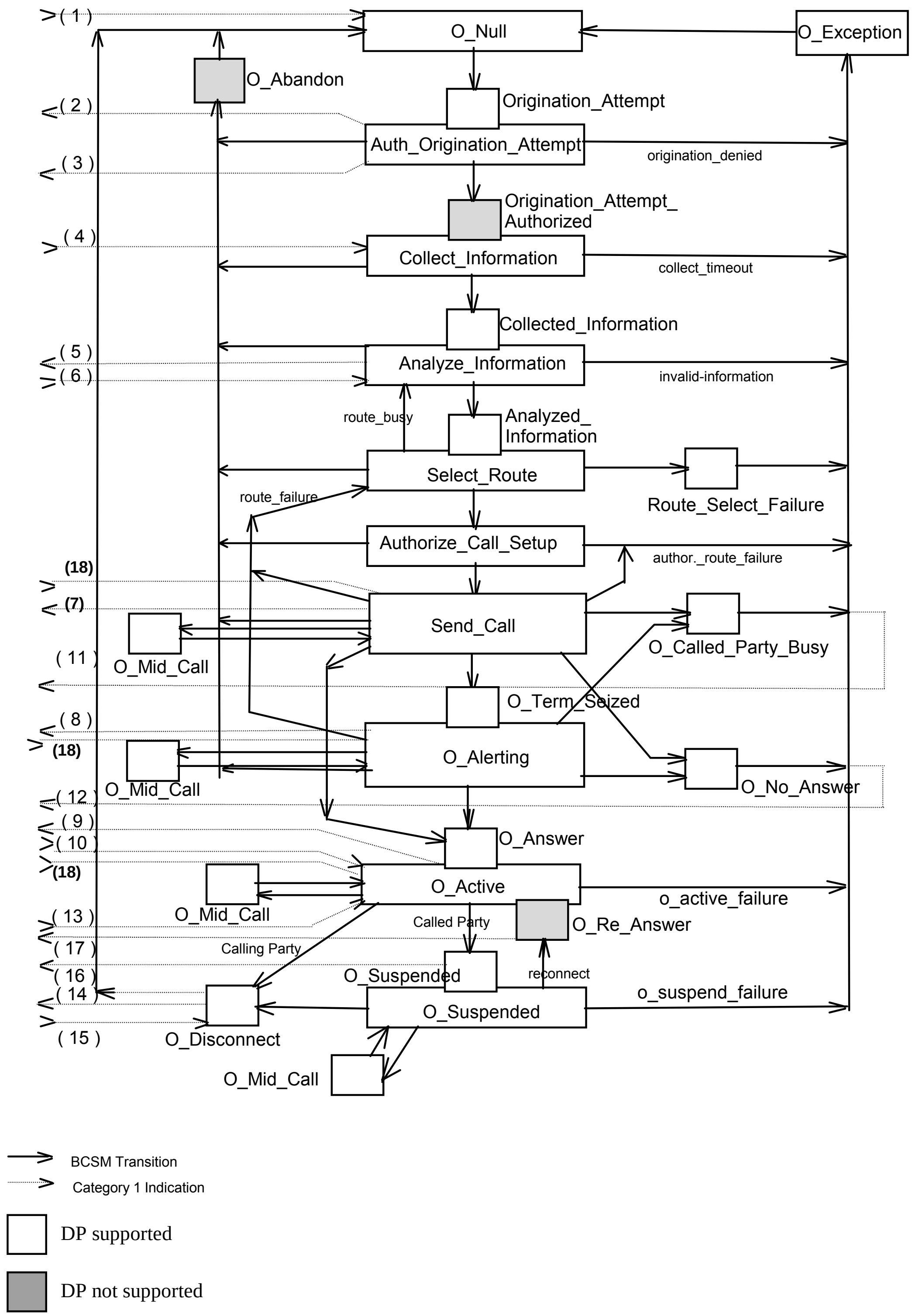


Figure 5 - User - O-BCSM access signaling indications for the T1 IN BCSM

NOTE - Indications that are shown as terminating on a DP in Figure 5 are received by the switch and are not part of the IN DP Processing.

5.3.3.2 T_BCSM - user access signaling indications (Category 2)

Definition: These Indications include the representation of the network's perception of possible actions taken by the called party as well as the called party's perception of actions taken by the network. The Indications are between a local exchange that is terminating a call and a user (i.e., called party). They include the definition of how actions by the terminating call model (user) affect the user (terminating call model). These Indications are derived from Access Signaling (e.g., DSS1, analog) as well as any other information that is available. Figure 6 illustrates these Indications.

Indications:

(1) An Indication is sent from T_BCSM to the User to terminate the call to an idle facility (e.g., SETUP).

(2) An Indication is sent from User to T_BCSM indicating that the User cannot accept the call (e.g., RELEASE COMPLETE).

(3) An Indication is sent from the User to the T_BCSM when the User determines compatibility with all call characteristics (e.g., SETUP ACKNOWLEDGE).

(4) The T_BCSM sends any remaining call information to the User (e.g., INFORMATION).

(5) An Indication is sent from the T_BCSM to the User upon the sending of sufficient call information.

(6) An Indication is sent from the User to the T_BCSM upon receipt of sufficient call information (e.g., CALL PROCEEDING).

(7) User sends an Indication to the T_BCSM that alerting is taking place (e.g., ALERTING).

(8) An Indication is sent from the User to the T_BCSM upon acceptance of the incoming call (e.g., CONNECT).

(9) An Indication is sent from the T_BCSM to the User acknowledging that the call can now be connected.

(10) An Indication is sent from the User to the T_BCSM that the User suspends the call.

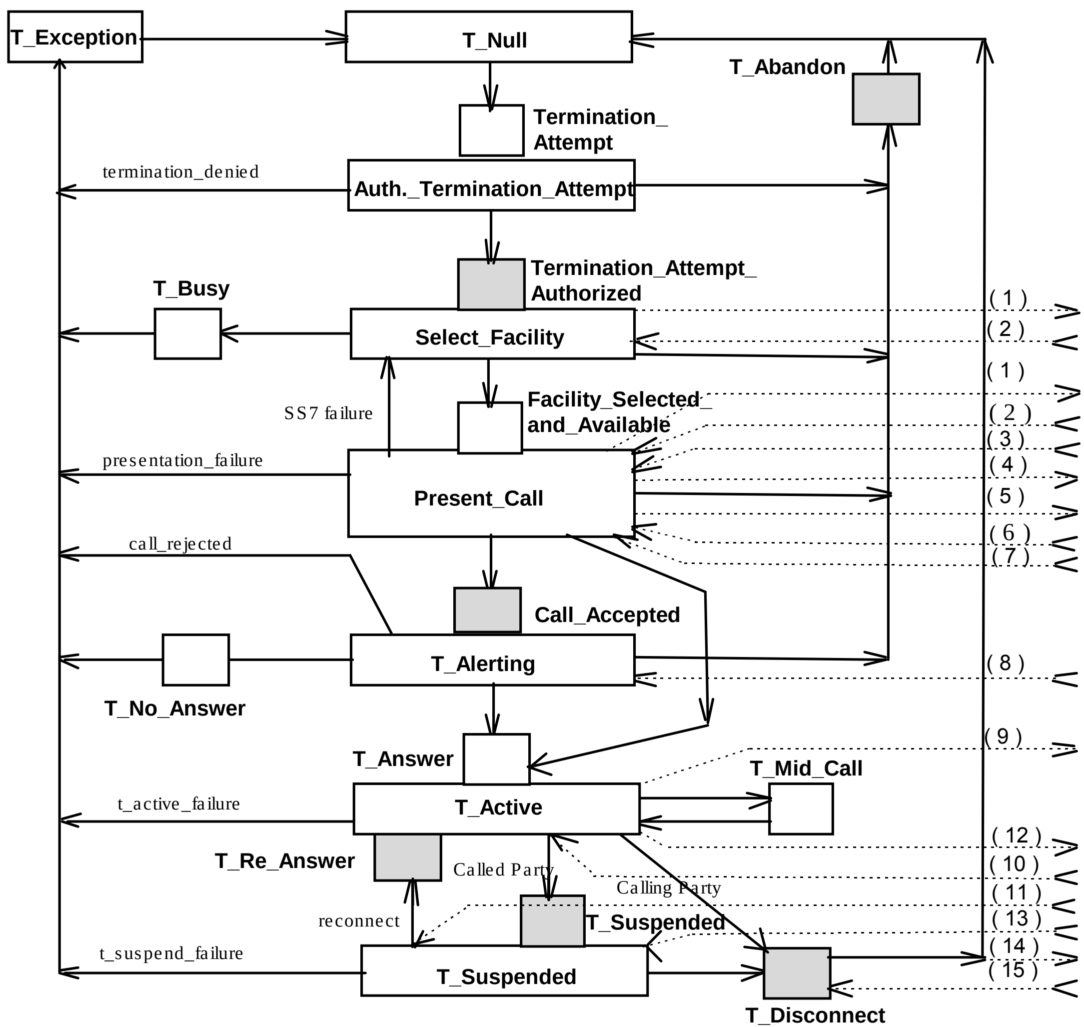
(11) An Indication is sent from the User to the T_BCSM that the User resumes the call.

(12) The T_BCSM sends an Indication to the User indicating that the calling party has gone on hook.

(13) An Indication is received by the T_BCSM from the User to end the call.

(14) The T_BCSM indicates to the User that the call is being disconnected.

(15) The User acknowledges to the T_BCSM that the call is being disconnected.



note 1 : Indications 1 and 2 at Select_Facility
for case of switch-CPE B channel negotiation

note 2 : Indications 1 and 2 at Present_Call
for case of no B channel negotiation

BCSM Transitions

Category 2 Indications

DP supported

DP not supported

Figure 6 - T-BCSM - User access signaling indications for the T1 IN BCSM

NOTE - Indications that are shown as terminating on a DP in Figure 6 are received by the switch and are not part of the IN DP processing.

5.3.3.3 Intra local exchange BCSM indications (Category 3)

Figure 7 illustrates the communication between two call segments in the SSF/CCF for a basic two-party call. It shows the indications that flow between the originating and terminating BCSMs for T1 IN. All possible indications are shown, except for any which may occur at the O_Exception and the T_Exception PICs. Note that these indications are not intended to be mapped to explicit protocol operations.

Explanations of indications concerning Figure 7:

(1) Initiate T_BCSM after the authority to place the call has been verified and a usable route has been identified. The O_BCSM is currently in the Send_Call PIC. The originating Basic Call Manager has sent the call attempt to the terminating Basic Call Manager for further processing.

(2) For SS7-supported trunks, if the received IAM indicates a continuity check is required and the resultant continuity check is successful, then an Indication is sent from the O_BCSM to the T_BCSM (causes T_Null PIC to Termination_Attempt DP BCSM transition in the T_BCSM).

(3) An Indication is sent from the T_BCSM to O_BCSM that the terminating line or trunk is busy (causes Send_Call PIC to O_Called_Party_Busy BCSM transition in O_BCSM, if terminating to a line. If terminating to a trunk, then this Indication causes Send_Call PIC to Select_Route PIC BCSM transition in the O_BCSM).

(4) An Indication is sent from the T_BCSM to O_BCSM that the terminating line or trunk is busy (causes O_Alerting PIC to O_Called_Party_Busy DP BCSM transition in O_BCSM).

(5) An Indication is sent from the T_BCSM to O_BCSM that the call cannot be presented (causes Send_Call PIC to Select_Route PIC, O_Called_Party_Busy DP, or O_No_Answer DP).

(6) An Indication is sent from the T_BCSM to the O_BCSM that an ISDN capable Called Party has signaled call acceptance with immediate BCSM transition to an answered (i.e., CONNect message) condition (causes Send_Call PIC to O_Answer DP BCSM transition in O_BCSM).

(7) An Indication is sent from T_BCSM to O_BCSM that Called Party is being alerted (causes O_BCSM to transit from Send_Call PIC O_Alerting PIC and prepare to send ring Indication to the Calling Party).

(8) An Indication is sent from T_BCSM to O_BCSM that Called Party has rejected the call (this is indicated to the O_BCSM with a busy cause and causes O_BCSM to transit from O_Alerting PIC to Select_Route PIC or O_Called_Party_Busy DP).

(9) An Indication is sent from T_BCSM to O_BCSM that Called Party has not answered within a specified time period (causes O_Alerting PIC to O_No_Answer DP BCSM transition in O_BCSM).

(10) An Indication is sent from the T_BCSM to the O_BCSM that called party has not answered within a specified time period (causes Send_Call PIC to O_No_Answer DP BCSM transition in O_BCSM).

(11) An Indication is sent from T_BCSM to O_BCSM that Called Party has accepted and answered the call attempt (causes O_Alerting PIC to O_Answer DP BCSM transition in O_BCSM).

(12) An Indication is sent from the T_BCSM to the O_BCSM that the called party has accepted and answered the call attempt (causes Send_Call PIC to O_Answer DP BCSM transition in O_BCSM).

(13) An Indication is sent from T_BCSM to O_BCSM that Called Party has disconnected (e.g., on hook) (causes O_Active PIC to O_Suspended DP BCSM transition in O_BCSM).

(14) An Indication is sent from T_BCSM to O_BCSM that Called Party re-answers is received before the timer expires (causes O_Suspended PIC to O_Re_Answer DP BCSM transition in O_BCSM).

(15) An Indication is sent from O_BCSM to T_BCSM that Calling Party has disconnected, while T_BCSM was in T_Active PIC (causes T_Active PIC to T_Disconnect DP BCSM transition in T_BCSM).

(16) An Indication is sent from O_BCSM to T_BCSM that Calling Party has disconnected, while T_BCSM was in T_Suspended PIC (causes T_Suspended PIC to T_Null PIC (T_Disconnect event) BCSM transition in T_BCSM).

(17) An Indication is sent from T_BCSM to O_BCSM that Called Party has disconnected (causes O_Suspended PIC to O_Disconnect DP BCSM transition in O_BCSM).

(18) An Indication is sent from the T_BCSM (T_Disconnect event) to O_BCSM that the calling party has disconnected (causes O_Active PIC to O_Disconnect DP BCSM transition in O_BCSM).

(19) An Indication is sent from O_BCSM to T_BCSM that Calling Party has abandoned (causes Authorize_Termination_Attempt PIC, Select_Facility PIC, Present_Call PIC or T_Alerting PIC to T_Null PIC (T_Abandon event) BCSM transition in T_BCSM).

NOTES

- Indications (15) and (17) are mutually exclusive;
- These Indications are for intra-switch;
- The Indications do not explicitly include the modeling of SRFs;
- Indications that are preceded by a DP may be affected depending on whether the DP is active and the SCF response.

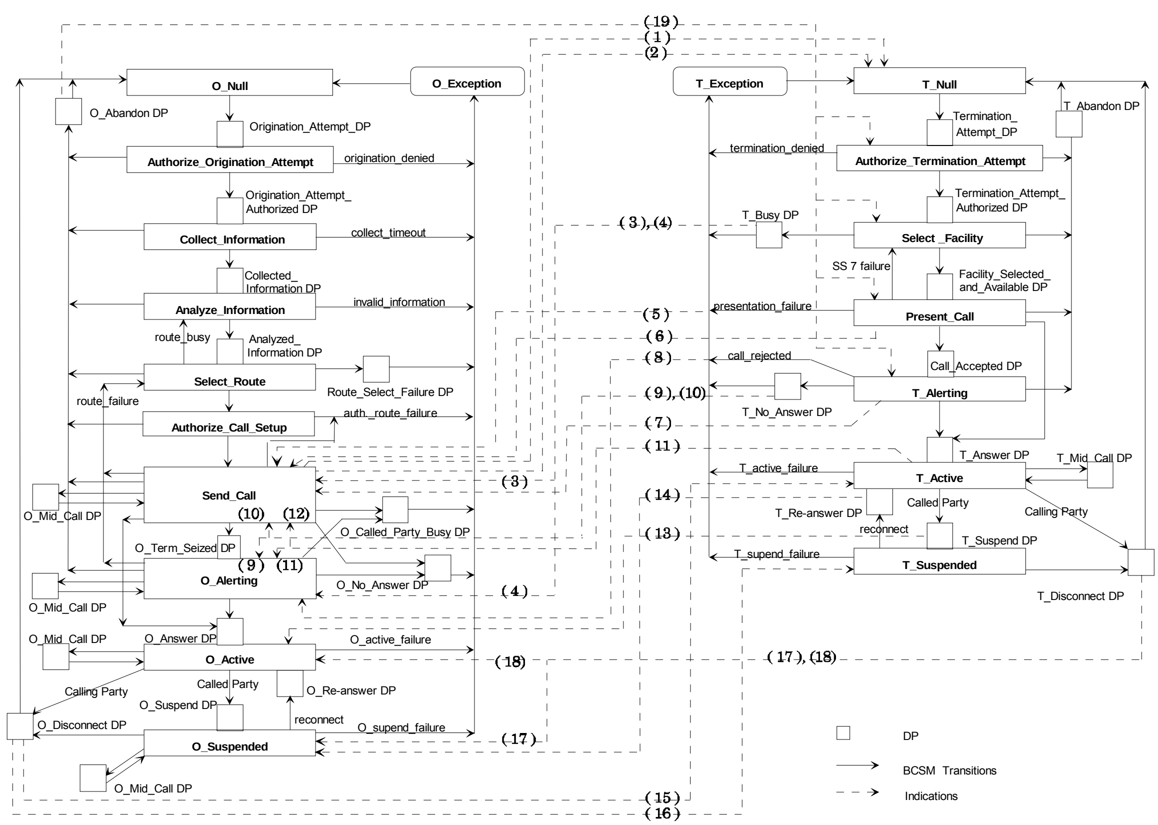


Figure 7 - BCSM indications

5.3.4 BCSM detection points

Certain basic call and connection events may be visible to IN service logic instances. DPs are the points in call processing at which these events are detected.

A DP can be armed in order to notify an IN service logic instance that the DP was encountered, and potentially to allow the IN service logic instance to influence subsequent call processing. If a DP is not armed, the SSF/CCF continues call processing without SCF involvement. DPs are characterized by the following four attributes:

- Arming/disarming mechanism – The mechanism by which the DP is armed. A DP may be statically armed or dynamically armed. A DP is statically armed through SMF service feature provisioning. A statically armed DP remains armed until explicitly disarmed by the SMF. A DP is dynamically armed by the SCF within the context of a call associated IN service control relationship. The following DP disarming rules apply:
  - If an armed EDP is met, then it is disarmed,
  - If an EDP is met that causes the release of the related leg, then all EDPs related to that leg are disarmed.
  - If a call is released, then all EDPs related to that call are disarmed.
- Criteria – In addition to the condition that a DP be armed, DP criteria are conditions that must be met in order to notify the SCF that the DP was encountered (see 5.3.5).
- Relationship – Given that an armed DP was encountered and DP criteria are met, the SSF may provide an information flow via a relationship:
  - If this relationship is between the SSF/CCF and the SCF for the purpose of call/service logic processing, it is considered to be an IN service relationship. This relationship may be of two types:
    - a control relationship if the SCF is able to influence call processing via the relationship;
    - a monitor relationship if the SCF is not able to influence call processing via the relationship.

    With respect to an IN service control relationship, the information flow provided by the SSF to the SCF on encountering a DP may initiate a control relationship, may be within the context of an existing control relationship, or may be within the context of an existing monitor relationship.
  - If this relationship is between the SSF/CCF and the SCF or SMF for management purposes, it is considered to be a service management control relationship.
- Call processing suspension – Given that an armed DP was encountered and DP criteria are met for an IN service control relationship, the SSF may suspend call processing to allow the SCF to influence subsequent call processing. When call processing is suspended, the SSF sends an information flow to the SCF requesting instructions, and waits for a response. When call processing is not suspended, the SSF sends an information flow notifying the SCF that a DP was encountered, and does not expect a response. This attribute is set by the same mechanism that arms the DP.

Based on these attributes, four types of DPs are identified for T1 IN. The DP types are:

- Trigger detection point – Request (TDP-R);
- Trigger detection point – Notification (TDP-N);
- Event detection point – Request (EDP-R);
- Event detection point – Notification (EDP-N).

These DP types are defined by the DP attribute values in Table 3.

BCSM DPs may be any one of these DP types. DP processing for each DP type is illustrated in Figure 8 and described in 5.3.7.

Table 3 - BCSM DP types

DP type	Arming mechanism	Criteria	IN service relationship	Suspension	Service feature examples
TDP-R	Static	Specific to DP	Initiates control relationship	Yes	All
TDP-N	Static	Specific to DP	Initiates and terminates monitor relationship	No	Televoting, call logging
EDP-R	Dynamic	None	Within context of existing control relationship	Yes	Call distribution, call rerouting distribution
EDP-N	Dynamic	None	Within context of existing control or monitor relationship	No	Charging for any service feature, call logging, call queuing

5.3.5 DP criteria

DP criteria are conditions that must be met in order to notify the SCF that the DP was encountered. These criteria can be assigned to a DP from the viewpoint of range of effectiveness, as identified below:

- Individual line/trunk based criteria

  This type of criteria applies to each subscriber line or trunk line. For example, SCF processing is invoked when user A makes call origination. This criterion could be said to be specific for user A.

- Group based criteria

  This type of criteria applies to a certain group of lines or users. For example, when a call origination from any user in a certain centrex group should invoke SCF processing the trigger should apply to that specific centrex group.

- Office based criteria

  This type of criteria applies to the whole office. Any calls generated in the switching system will be subject to these criteria. For example, any call that makes access to the registered Freephone number is triggered and SCF processing is invoked.

The following criteria are DP criteria for T1 IN, as applicable for a given DP:

- trigger assigned (unconditional/conditional on other criteria);

- Class of service;
- specific B-channel identifier;
- specific digit strings;
- feature codes (e.g., *XX, #);
- prefixes (e.g., 0+, 00+, 0–, 00–, 011, 01, 1+);
- access codes (e.g., 8+) for customized numbering plan;
- specific abbreviated dialing strings for customized numbering plan;
- specific calling party number strings;
- specific *Called Party Number* strings;
- nature of address (e.g., subscriber significant number, national significant number, international number);
- bearer capability;
- feature activation/indication (unconditional/conditional on specific feature patterns);
- facility information (unconditional/conditional on specific facility information patterns);
- cause (unconditional/conditional on specific cause patterns).

With respect to the DP criteria listed above, note that these DP criteria only apply to TDPs. DP criteria for Event Detection Points (EDPs) are addressed by the RequestReportBCMEvent operation. In addition, note that one or more DP criteria may be applicable at a given DP. The assignment of DP criteria to a TDP and the combinations of DP criteria applicable at a given DP continue to evolve. Further DP criteria and specific assignment of DP criteria to TDPs/EDPs may evolve through future versions. Note further that the assignment of DP criteria to a TDP on either a line/trunk, group or office basis may have an impact on the memory and real-time performance requirements of the SSF/CCF. The DP criteria for T1 IN are defined below, as applicable to a given TDP.

Note that the applicability of DP criteria at a given DP depends on when call processing information is available and how long it is retained. If network and service providers plan to implement T1 IN services in a multi-supplier environment, they should consider formulating such requirements to ensure consistent implementations across supplier equipment. Such requirements should be considered carefully so as not to adversely impact memory and real-time performance aspects of SSF/CCF processing.

1) *Trigger assigned* (unconditional/conditional on other criteria) - An indicator of the armed/disarmed status of a TDP assigned on a line/trunk, group, or office basis.

   The trigger assigned criterion can be used by itself or in conjunction with other criteria at a TDP. If the trigger assigned criterion is unconditional at a TDP, then it is used by itself - no other DP criteria need to be satisfied at the TDP before informing the SCF that the TDP was encountered. If the trigger assigned criterion is conditional at a TDP, then it is used in combination with other criteria at the TDP - - all of the other DP criteria in the combination need to be satisfied before informing the SCF that the TDP was encountered.

   Applies at all DPs (all DPs can be provisioned as TDPs).

2) *Class of Service* - This is either a (i) customer class of service, (ii) trunk class of service, or (iii) private facility class of service; (i) is a code that identifies all attributes of a line that require distinctive call processing treatment (e.g., for party lines and coin lines), (ii) is a code that identifies attributes of a trunk group such as type of signaling used, and (iii) is a code that identifies attributes of a private trunk group such as type of signaling used and flash repeat capability.
   Originating access (user/network) class of service is available at the Origination_Attempt DP and could be applicable at any of the originating DPs. Terminating access (user/network)

class of service is available at the Termination_Attempt DP and could be applicable at any of the terminating DPs.

3) *Specific B-channel identifier* - An identifier of the specific B-channel on an ISDN interface from which a call attempt has originated or to which a call attempt is to be terminated.

A-party B-channel identifier is available at the Origination_Attempt DP for a party served by an ISDN interface only and could be applicable all originating DPs. B-party B-channel identifier is available during the Select_Facility PIC after an idle terminating facility has been selected for a party served by an ISDN interface only and could be applicable at the Facility_Selected_and_Available, T_No_Answer, T_Answer, T_Mid_Call, and at T_Abandon (only after an idle terminating facility has been selected).

4) *Specific digit strings* - A string of digits that must match collected digit strings for numbering plans in which a variable number of digits are to be collected. It could be zero or more digits (e.g., to trigger on "off-hook delay").

The string of digits should be consistent with the structure of the dialing plan and should be administrable. For example, the network provider may specify the first N digits where N is consistent with the structure of the ITU-T Recommendation E.164 numbering plan, or any other appropriate numbering plan.

Collected digit strings can be available at the Origination_Attempt DP for a party served by an ISDN interface using en bloc sending and at the Collected_Information DP for a party served by a non-ISDN line. Since collected digit strings are not analyzed until the Analyze_Information PIC (except to determine if a sufficient number of digits have been collected), this criterion could be applicable at the Analyzed_Information DP and beyond. The Analyzed_Information DP [mandatory] and all those originating DPs that may be encountered after Analyzed_Information [optional] are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

- collected digit string can be available at the Origination_Attempt DP through ISUP signaling for an SS7 trunk.
- collected digit string can be available at the Collected_Information DP for a party served by a conventional trunk (e.g., non-SS7), ISDN interface using overlap sending, and private facilities.

5) *Feature codes* (e.g., *XX, #) - A vertical service code, such as a "#" or a two-digit or three-digit code preceded by "*" or "11," that precedes any subsequent digit collection (e.g., according to the "normal numbering plan").

Feature codes can be available at the Origination_Attempt DP for a party served by an ISDN interface using en bloc sending or through ISUP signaling for an SS7 trunk, and can be available at the Collected_Information DP for non-ISDN lines and private facilities. Since collected digit strings are not analyzed until the Analyze_Information PIC (except to determine if sufficient information has been collected), this criterion could be applicable at the Analyzed_Information DP and beyond. The Analyzed_Information DP [mandatory] and all those originating DPs that may be encountered after Analyzed_Information [optional] are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Feature codes can be available at the Collected_Information DP for a party served by an ISDN interface using overlap sending.

6) *Prefixes* (e.g., 0+, 00+, 011, 01, 1+) - A string of digits that are not feature codes or access codes and which precede any subsequent digit collection (e.g., according to the "normal numbering plan").

Prefixes can be available at the Origination_Attempt DP for a party served by an ISDN interface using en bloc sending, and can be available at the Collected_Information DP for non-ISDN lines, conventional trunks, and private facilities. Since collected prefix information is not analyzed until the Analyze_Information PIC (except to determine if sufficient information has been collected), this criterion could be applicable at the Analyzed_Information DP and beyond. The Analyzed_Information DP [mandatory] and all those originating DPs that may be encountered after Analyzed_Information [optional] are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Prefixes can be available at the Collected_Information DP for a party served by an ISDN interface using overlap sending.

7) *Access codes* (e.g., 8+) for customized numbering plan - A string of digits in a customized numbering plan that matches access codes such as attendant access codes, access codes to escape to the public network, access codes to access a private facility, access codes to access a private network, and feature access codes.

Access codes can be available at the Origination_Attempt DP for a party served by an ISDN interface using en bloc sending, and can be available at the Collected_Information DP for non-ISDN lines and private facilities. Since collected access codes are not analyzed until the Analyze_Information PIC (except to determine if sufficient information has been collected), this criterion could be applicable at the Analyzed_Information DP and beyond. The Analyzed_Information DP [mandatory] and all those originating DPs that may be encountered after Analyzed_Information [optional] are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Access codes can be available at the Collected_Information DP for a party served by an ISDN interface using overlap sending.

8) *Specific abbreviated dialing strings* for customized numbering plan - An abbreviated *Called Party Number* in a customized numbering plan that must match collected address information.

Abbreviated address information can be available at the Origination_Attempt DP for a party served by an ISDN interface using en bloc sending, and at the Collected_Information DP for a party served by a non-ISDN line or private facilities. Since collected address information is not analyzed until the Analyze_Information PIC (except to determine if sufficient information has been collected), this criterion could be applicable at the Analyzed_Information DP and beyond. The Analyzed_Information DP [mandatory] and all those originating DPs that may be encountered after Analyzed_Information [optional] are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Specific abbreviated dialing strings can be available at the Collected_Information DP for a party served by an ISDN interface using overlap sending.

9) *Specific calling party number strings* - A string of digits that must match the calling party number, which is a local, national, or international ITU-T Recommendation E.164 number or a number in a customized numbering plan. If a call has been forwarded, the calling party number is the number of the original calling party.

The calling party number is available at the Origination_Attempt DP in the originating BCSM and the Termination_Attempt DP in the terminating BCSM for a call originating from a non-ISDN line, ISDN interface, and can be available at the Origination_Attempt DP and the Termination_Attempt DP for SS7 trunks. This criterion could be applicable at all DPs.

10) *Specific called party number strings* - A string of digits that must match the *Called Party Number*, which is either a local, national, or international ITU-T Recommendation E.164 number, or a number in a customized numbering plan; the latter is not supported by SS7 or conventional trunks. If a call has been forwarded, the *Called Party Number* is the number of the party to which the call is forwarded.

The *Called Party Number* can be available at the Origination_Attempt DP for a party served by an ISDN interface using en bloc sending or for an SS7 trunk, and can be available at the

Collected_Information DP otherwise. Since collected address information is not analyzed until the Analyze_Information PIC (except to determine if sufficient information has been collected), this criterion could be applicable at the Analyzed_Information DP and beyond, and at all terminating DPs. In the originating BCSM, the Analyzed_Information DP [mandatory] and all those originating DPs that may be encountered after Analyzed_Information [optional] are proposed. No specific proposals are made for the DPs in the terminating BCSM.

11) *Nature of address* (e.g., Subscriber Significant Number, National Significant Number, International Number) - An indicator of whether the *Called Party Number* is a private, local (or subscriber), national, or international number

The nature of address is available at the Analyzed_Information DP. This criterion could be applicable at the Analyzed_Information DP and beyond. The Analyzed_Information DP [mandatory] and all those originating DPs that may be encountered after Analyzed_Information [optional] are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

12) *Bearer capability* - The bearer capability information is available at the Origination_Attempt DP. This criterion could be applicable at all DPs.

13) *Feature activation/indication* (unconditional/conditional on specific feature patterns) - In a local exchange only, a feature activation/indication on an ISDN interface or that is detected at the Mid_Call DP (e.g., “hook-flash,” #, etc.) for ISDN and non-ISDN lines that can be sent in conjunction with or preceding other address/digit collection.

A feature activation/indication can be available at all DPs in the originating BCSM for a party served by an ISDN interface, and can be available at the O_Mid_Call DP in the originating BCSM for a party served by a non-ISDN line. A feature activation/indication can be available at the T_No_Answer, T_Answer, and T_Mid_Call DPs in the terminating BCSM for a party served by an ISDN interface, and can be available at the T_Mid_Call DP in the terminating BCSM for a party served by a non-ISDN line. Since collected feature activation information is not acted upon before the Analyze_Information PIC in the originating BCSM and before the T_Alerting PIC in the terminating BCSM, this criterion could be applicable at the Analyzed_Information DP and beyond, and at the following terminating DPs: T_No_Answer, T_Answer, and T_Mid_Call.

14) *Facility information* (unconditional/conditional on specific facility information patterns) - A match on the Facility Information Element contained in a signaling message as defined in DSS1 and ISUP.

15) *Cause* (unconditional/conditional on specific cause patterns) - A match on the Cause IE contained in a signaling message as defined in DSS1 and ISUP or an indicator of the cause of specific events of interest.

Route selection failure information is available at the Route_Select_Failure DP, busy cause information is available at the O_Called_Party_Busy and T_Busy DPs, and release cause information is available at the O_Disconnect DP. This criterion is applicable at the identified DPs.

DP criteria assignment to a TDP is dependent on the information available at that TDP and the information available at a TDP is described in 5.3.

Table 4 (Originating DP criteria) and Table 5 (Terminating DP criteria) denote the applicability of DP criteria to all the DPs.

The entries in the table can be:

- Customer Based
- Trunk group Based

- Private Facility Based
- Office Based

The DPs in Table 4 and Table 5 are abbreviated as follows:

Originating DPs			Terminating DPs		
OA	=	Origination_Attempt	TA	=	Termination_Attempt
CI	=	Collected_Information	TB	=	T_Busy
AI	=	Analyzed_Information	FSA	=	Facility_Selected_and_Available
RSF	=	Route_Select_Failure	TNA	=	T_No_Answer
OCPB	=	O_Called_Party_Busy	TAns	=	T_Answer
ONA	=	O_No_Answer	TMC	=	T_Mid_Call
OAns	=	O_Answer			
OMC	=	O_Mid_Call			
OS	=	O_Suspended			
OD	=	O_Disconnect			

Table 4 - Originating DP criteria

	Originating DP									
DP criterion	**OA**	**CI**	**AI**	**RSF**	**OCPB**	**ONA**	**OAns**	**OMC**	**OS**	**OD**
Trigger Assigned	X	X	X	X	X	X	X	X	X	X
Class of Service	X	O	O	O	O	O	O	O	O	O
Specific Calling Party Number (NOTE 4)	X	O	O	O	O	O	O	O	O	O
Bearer Capability	X	O	O	O	O	O	O	O	O	O
Specific B-channel Identifier	O	O	O	O	O	O	O	O	O	O
Specific Digit String (NOTE 1)	-	X	X	O	O	O	O	O	O	O
Feature Code (NOTE 1)	-	X	X	O	O	O	O	O	O	O
Prefixes (NOTE 1)	-	X	X	O	O	O	O	O	O	O
Access Codes (NOTE 1)	-	X	X	O	O	O	O	O	O	O
Called Party Number (NOTE 1)	-	X	X	O	O	O	O	O	O	O
Specific abbreviated dialing string (NOTE 1)	-	-	X	O	O	O	O	O	O	O
Nature of Address	-	-	X	O	O	O	O	O	O	O
Feature Activation (NOTE 3)	-	-	X	X	X	X	X	X	X	X
Facility Information (NOTE 2)	-	-	X	-	-	-	X	X	-	-
Cause	-	-	-	X	X	-	-	-	-	X

X : Applicable **- : Not Applicable** **O : Optional**

NOTE 1: Same type of trigger requiring analysis of a specific number of received digits. The analysis can be based on the complete number of received digits or can be based on a predefined number of digits starting from the most significant digit of the received information.

NOTE 2: A match on the Facility Information Element contained in a signaling message as defined in DSS1 and ISUP.

NOTE 3: In a local exchange only. The BCSM has to analyze (if facility is allowed, stored as Class of Service attribute) the received information and has to initiate an IN trigger if required. A feature activation/indication can be available at all DPs in the originating BCSM for a party served by an ISDN interface and can be available at the O_Mid_Call DP in the originating BCSM for a party served by a non-ISDN line. A feature activation/indication can be available at the T_No_Answer, T_Answer, and T_Mid_Call DPs in the terminating BCSM for a party served by an ISDN interface and can be available at the T_Mid_Call DP in the terminating BCSM for a party served by a non-ISDN line.

NOTE 4: The analysis should not be based on the complete calling party number, it shall be based on a predefined number of digits, starting from the most significant digit of the calling party number.

Table 5 - Terminating DP criteria

	Terminating DP					
DP criterion	**TA**	**TB**	**FSA**	**TNA**	**TAns**	**TMC**
Trigger Assigned	X	X	X	X	X	X
Class of Service	X	O	O	O	O	O
Specific Calling Party Number (NOTE 4)	X	O	O	O	O	O
Bearer Capability (NOTE 5)	O	O	O	O	O	O
Specific B-channel Identifier	-	-	O	O	O	O
Specific Digit String (NOTE 1)	-	-	-	-	-	-
Feature Code (NOTE 1)	-	-	-	-	-	-
Prefixes (NOTE 1)	-	-	-	-	-	-
Access Codes (NOTE 1)	-	-	-	-	-	-
Called Party Number (NOTE 1)	-	-	-	-	-	-
Specific abbreviated dialing string (NOTE 1)	-	-	-	-	-	-
Nature of Address	-	-	-	-	-	-
Feature Activation (NOTE 3)	-	-	-	X	X	X
Facility Information (NOTE 2)	-	-	-	-	X	X
Cause	-	X	-	-	-	-

X : Applicable **- : Not Applicable** **O : Optional**

NOTE 1: Same type of trigger requiring analysis of a specific number of received digits. The analysis can be based on the complete number of received digits or can be based on a predefined number of digits starting from the most significant digit of the received information.

NOTE 2: A match on the Facility Information Element contained in a signaling message as defined in DSS1 and ISUP.

NOTE 3: In a local exchange only. The BCSM has to analyze (if facility is allowed, stored as Class of Service attribute) the received information and has to initiate an IN trigger if required. A feature activation/indication can be available at all DPs in the originating BCSM for a party served by an ISDN interface and can be available at the O_Mid_Call DP in the originating BCSM for a party served by a non-ISDN line. A feature activation/indication can be available at the T_No_Answer, T_Answer, and T_Mid_Call DPs in the terminating BCSM for a party served by an ISDN interface and can be available at the T_Mid_Call DP in the terminating BCSM for a party served by a non-ISDN line.

NOTE 4: The analysis should not be based on the complete calling party number, it shall be based on a predefined number of digits, starting from the most significant digit of the calling party number.

If a criterion is marked with an “X” for a Detection Point, then this means that a conditional TDP that is armed at the Detection Point may require the criterion as listed in the table to be satisfied before informing the SCF that the TDP was encountered. e.g., a conditional TDP at the Origination_Attempt DP may require the class of service criterion to be satisfied before the SCF is informed that the TDP was encountered.

If a criterion is marked with an "O" for a Detection Point, then this means that it is implementation dependent if the criterion specific information is still present at that DP because not all suppliers may retain this information for the duration of the call/attempt. If the information is still present, the treatment is the same as a criterion marked with an "X".

The trigger item is defined as a single set of DP criteria and the associated information that an SSF/CCF uses to determine if the criteria are met and how to process the trigger. The trigger item consists of trigger type, DP criteria, and the SCF routing information. The trigger items are assigned to users by management process. An SSF should use the SCF routing information to format and route the messages to the appropriate SCF application. The SCF may use existing MTP/SCCP capabilities to route to the SCF.

5.3.6 Trigger types and trigger precedence

Trigger types denote classes of events of interest. They are used to establish trigger precedence rules at TDPs and indicate to the SCF the service logic to be invoked. A non-exhaustive list of trigger types is defined. Implementation of this set of trigger types, or a subset of this set of trigger types, and further network provider defined trigger types are defined by the network operator. This clause describes the current set of trigger types defined. The trigger types given are described in terms of:

(1) TDP - The TDP at which the trigger can be detected.

(2) DP criteria - The conditions needed to trigger.

(3) Category - Office, group, or subscribed (line based).

(4) Interface - Type of interface to which it can be assigned (e.g., ISDN line).

(5) Trigger Type - The value that identifies the type of criteria that caused the SSF/CCF to detect a valid trigger condition at this TDP (i.e., the trigger type).

(6) Fault handling - Defines fault handling procedures for the case when the SCF does not respond to the SSF/CCF message.

Escape codes apply to the Off-Hook_Delay, Channel_Setup_PRI, Shared_Interoffice_Trunk and Specific_Digit_String triggers. These codes provide an escape so that a subscriber to these triggers can still make certain calls (e.g., Emergency 911) when the SCF or the link to the SCF is down. Escape codes also allow more flexibility in the provisioning of triggers. For example, if a Specific_Digit_String trigger is provisioned with a 6-digit criterion, then it may be desirable to escape triggering for a particular 10-digit sequence (e.g., 215-555-1212) that matches the 6-digit criterion (e.g., 215-555).

Each SSF/CCF will have an administered list of escape codes that apply to all Off-Hook_Delay and Channel_Setup_PRI triggers in the SSF/CCF. A separate administered list of escape codes in each SSF/CCF will apply to all Shared_Interoffice_Trunk triggers in the SSF/CCF. Each SSF/CCF shall also support an administered list of escape codes for each Specific_Digit_String trigger in the SSF/CCF. Escape codes are valid numbers as per the number plan, prefixed numbers (e.g., 0-, 00-), or an Emergency Service Code. The SSF/CCF provides the same escape treatment to calls in the following situations:

- A call from a line or trunk using the numbering plan in force corresponds to a number on the administered escape code list.
- A call from a line or trunk using the numbering plan in force corresponds to a prefix and number on the escape code list.
- A call from a line or trunk using the numbering plan in force corresponds to a feature code plus a number on the escape code list.

- A call from a line or trunk using a private numbering plan corresponds to an access code that results in the call being routed over the public network plus a number on the escape code list.

Network administrators should be aware that if a number is on the escape code list, then variations of that number that may occur because of different user dialing procedures may also need to be entered on the escape code list so that they also result in escape code treatment. If the SSF/CCF determines that an escape code applies, then the Off-Hook_Delay trigger is not detected, a message is not sent to the SCF from the Collected_Information TDP, and the SSF/CCF continues with normal call processing at the Analyze_Information PIC. Triggers may be detected at subsequent TDPs.

After the BRI feature activator, public feature code, or customized numbering plan trigger is encountered, the SSF/CCF supports subsequent digit collection. Subsequent digit collection rules are the same for all subscribers to a particular numbering plan. This subsequent digit collection is specified by the numbering plan in force and, thus, is assumed to occur during the Collect_Information PIC. Any subsequent digits are included in an CollectedInformation query if an Off_Hook_Delay trigger is hit.

The SSF/CCF supports the administration of the following rules for subsequent digit collection for a numbering plan, not an individual subscriber trigger. Each BRI feature activator, public feature code, or access code is administrable to do one of the following:

- Do not collect subsequent digits.
- Collect subsequent digits according to the normal numbering plan for that line or trunk. Second dial tone is applied after the feature code or indicator is dialed. A restart clears the digits dialed after the feature code, but does not clear the feature code itself. If the digits dialed are incomplete or invalid, the SSF/CCF shall not query the SCF, but will provide final treatment.
- Collect a variable number of digits (0 to 32). Digit collection is complete when the caller enters a “#” to indicate end of dialing, or when the normal interdigit timing interval expires.

As the same DP may be armed multiple times as a TDP-R, precedence rules for trigger processing are specified as:

- For ISDN BRI lines, the same trigger may be assigned to the ISDN line interface, the ISDN Service Profile, or the Directory Number and Call Type (DN/CT). These may be provisioned with different SCFs as the destination. The sequence of processing shall be DN/CT first, followed by Service Profile, and ISDN line last.
- Subscribed triggers have precedence over group triggers.
- Group triggers have precedence over office triggers.

The sequence for processing T1 IN triggers should be as detailed in Table 6. Table 6 reflects the above trigger precedence rules, giving higher precedence triggers first within the TDPs.

Table 6 - T1 IN trigger precedence

Trigger Detection Point	Trigger Type
Collected_Information	Dedicated_Trunk_Group
	Off-Hook_Delay
	Channel_Setup_PRI
	Shared_Interoffice_Trunk
Analyzed_Information	BRI_Feature_Activation_Indicator
	Specific_Feature_Code
	Public_Feature_Code
	Customized_Dialing_Plan
	Specific_Digit_String
	Emergency_Service

5.3.6.1 Off-Hook_Immediate

The SSF/CCF detects the Off-Hook_Immediate trigger when an origination indication from the interface is detected. The Origination_Attempt TDP has been reached.

1. TDP: Origination_Attempt
2. DP criterion: Trigger Assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group, Directory Number and Call Type.
5. Trigger Type: offHookImmediate
6. Fault Handling: Final treatment

5.3.6.2 Dedicated_Trunk_Group

The Dedicated_Trunk_Group trigger type denotes an event class on the originating interface whenever that the SSF/CCF receives a call over a trunk group that is marked in translations as a dedicated trunk group. If the SSF/CCF encounters a Dedicated_Trunk_Group trigger, then the SSF/CCF shall send a CollectedInformation TDP-Request message to the SCF (See T1.628 for Emergency Calling Service description and usage).

1. TDP: Collected_Information
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed (on trunk basis)
4. Trigger assigned to: Public trunk groups
5. Trigger Type: dedicatedTrunkGroup
6. Fault handling: Network Specific

5.3.6.3 Off-Hook_Delay

The Off-Hook_Delay trigger type denotes an event class on the originating interface such that the SSF/CCF receives enough information to process the call, information received does not violate the numbering plan in force, and escape codes/switch-based feature codes that prevent the Off-Hook_Delay trigger from being detected have not been entered. The Collected_Information TDP has been reached. This trigger occurs for all calls, but only on call origination. For example, the Off-Hook_Delay trigger type may be used in providing a feature to request user PIN prior to authorization of routing the call, after dialing is complete.

1. TDP: Collected_Information
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, BRI Service Profile, BRI interface, PRI interface, private facility trunk group, Directory Number and Call Type
5. Trigger Type: offHookDelay
6. Fault handling: Final Treatment

5.3.6.4 Channel_Setup_PRI

The Channel_Setup_PRI trigger type denotes a call attempt on a specific B-channel on a PRI interface. This trigger is detected when SSF/CCF receives enough information to process the call, information received does not violate the numbering plan in force, and escape codes/switch-based feature codes that prevent the Channel_Setup_PRI trigger from being detected have not been entered. This trigger occurs for all calls that use identified B-channels. For example, if a B-channel is dedicated to a specific service (e.g., directory assistance), this trigger could be used in providing this service on the B-channel.

1. TDP: Collected_Information
2. DP criteria: Trigger assigned (conditional), Specific B-channel identifier
3. Category: Subscribed
4. Trigger assigned to: Dedicated B-channel on a PRI interface
5. Trigger Type: channelSetupPRI
6. Fault handling: Final Treatment

5.3.6.5 Shared_Interoffice_Trunk

The Shared_Interoffice_Trunk trigger type denotes a class of events in which an SSF/CCF performs trigger processing for a CCF. This trigger is detected for calls routed to an SSF/CCF from a CCF: an assist procedure is to be invoked. The SSF/CCF supports a shared inter-office trunk trigger for the interface to a CCF. On this interface, SS7 or conventional signaling used by the CCF to route the call is used by the SSF/CCF for triggering a query to the SCF. For example, this trigger may be used in providing a feature to supply specific routing of a call when an assist has been invoked.

1. TDP: Collected_Information
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed (on trunk basis)

4. Trigger assigned to: Public trunk groups
5. Trigger Type: sharedIOTrunk
6. Fault handling: Final Treatment

5.3.6.6 BRI_Feature_Activation_Indicator

The BRI_Feature_Activation_Indicator trigger type is detected when the SSF/CCF detects a feature activation indicator. If the SSF/CCF receives a feature activator for a switch-based feature with or without dialed digits on the ISDN BRI line, it is desirable that the call escapes the Off-Hook_Delay trigger even if the other requirements for the trigger are met. This trigger is used for BRI features.

1. TDP: Analyzed_Information
2. DP criteria: Trigger assigned (conditional), BRI feature activation/indication
3. Category: Subscribed
4. Trigger assigned to: BRI Service Profile
5. Trigger Type: featureActivator
6. Fault handling: Final Treatment

5.3.6.7 Public_Feature_Code

The Public_Feature_Code trigger type is detected when any of the SSF/CCF supported feature codes (e.g., *XX) are dialed. Although several feature codes may cause the trigger to be detected, the SSF/CCF shall not differentiate between them for subscription purposes (e.g., *46, *53, *58 all cause the trigger to be detected). It is desirable that calls with a feature code that correspond to the activation, deactivation, or access to a switch-based feature with or without dialed digits for a call escape the Off-Hook_Delay trigger even if the other requirements for the trigger are met. For example, this trigger may be used within a service provider’s network to advertise other capabilities to the user when an existing feature code is used.

1. TDP: Analyzed_Information
2. DP criteria: Trigger assigned (conditional), Feature Code (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, BRI Service Profile, BRI interface, Directory Number and Call Type
5. Trigger Type: verticalServiceCode
6. Fault handling: Final Treatment

5.3.6.8 Specific_Feature_Code

The Specific_Feature_Code trigger type is detected when the SSF/CCF analyzes a specific feature code that is administered as a DP criterion (e.g., only *46 causes the trigger to be detected). It is desirable that calls with a feature code that corresponds to the activation, deactivation, or access to a switch-based feature with or without dialed digits for a call escape the Off-Hook_Delay trigger even if the other requirements for the trigger are met. Invocation of call forward can be supported by this trigger type.

1. TDP: Analyzed_Information

2. DP criteria: Trigger assigned (conditional), feature code on specific feature pattern
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, BRI Service Profile, BRI interface, Directory Number and Call Type
5. Trigger Type: specificFeatureCode
6. Fault handling: Final Treatment

5.3.6.9 Customized_Dialing_Plan

The Customized_Dialing_Plan trigger type is detected when an access code is dialed within a customized numbering plan. For example, 8+ or a one- to seven-digit intercom code can be defined as a trigger to query the SCF. The customized numbering plan specifies, for each code, whether the SSF/CCF performs some manipulation of the dialed digits (e.g., digit insertion, deletion, or translation to public numbers), or queries the SCF with the digits as dialed. It is desirable that calls from users of a customized numbering plan with a feature access code that corresponds to the activation, deactivation, or access to a switch-based feature with or without dialed digits escapes the Off-Hook_Delay trigger even if the other requirements for the trigger are met. For example, this trigger type could be used in providing 5-digit dialing through the public network for a virtual private network.

1. TDP: Analyzed_Information
2. DP criteria: Trigger assigned (conditional), Access Code or specific abbreviated dialing string for customized numbering plan
3. Category: Group
4. Trigger assigned to: All lines and trunks assigned to a customized numbering plan or business group
5. Trigger Type: customizedAccess, customizedIntercom
6. Fault handling: Final Treatment

5.3.6.10 Specific_Digit_String

The Specific_Digit_String trigger type is detected when the appropriate sequence of digits is dialed according to the numbering plan in use. For example, a 3-, 6-, or 10-digit sequence of digits can be provisioned as the trigger. Trigger provisioning specifies whether the SSF/CCF performs some manipulation of the dialed digits (e.g., digit insertion, deletion, or translation to public numbers), or queries the SCF with the digits as dialed. The numbering plan in force should ensure that emergency service numbers are distinct from provisionable specific digit strings. Precedence should be specified (e.g., most to least specific): for further study. For example, this trigger could be used to provide for customized call routing on a specific directory number.

1. TDP: Analyzed_Information
2. DP criteria: Trigger assigned (conditional), Specific *Called Party Number* string
3. Category: Office
4. Trigger assigned to: All lines and trunks assigned to a public office numbering plan or the entire SSF/CCF (i.e., not every number may have facilities associated with it).
5. Trigger Type: As per numbering plan (e.g., International ITU-T Recommendation E.164, National, private numbering plans)

6. Fault handling: Final Treatment

5.3.6.11 Emergency Service

The Emergency_Service trigger type is detected when a digit string denoting emergency service is dialed. The SSF/CCF detects the Emergency_Service trigger on any call with access to the public office numbering plan when a designated Emergency Service (e.g., 911) number is dialed. This trigger can provide for emergency call handling under control of the SCF.

1. TDP: Analyzed_Information
2. DP criteria: Trigger assigned (conditional), Specific *Called Party Number* string
3. Category: Office
4. Trigger assigned to: All lines and trunks assigned to a public office numbering plan or the entire SSF/CCF.
5. Trigger Type: emergencyService
6. Fault handling: Final Treatment

5.3.6.12 AFR

The AFR trigger type denotes exhaustion of a list of selected routes within the SSF/CCF. This is used to specify route choices for both public and private network calls. A specific list of routes is assigned to each user. The SSF/CCF detects the Automatic Flexible Routing (AFR) trigger when the list of routes is exhausted (i.e., all routes are busy or unavailable). For example, this trigger type could be used in providing a feature to play a customized announcement when all routes are busy.

1. TDP: Route_Select_Failure
2. DP criterion: Trigger assigned (conditional)
3. Category: Group
4. Trigger assigned to: All lines and trunks with access to the AFR pattern
5. Trigger Type: aFR
6. Fault handling: Final Treatment

5.3.6.13 O_Called_Party_Busy

The SSF/CCF detects the O_Called_Party_Busy trigger when the originating call portion receives a report of user busy from the terminating call portion. For example, this trigger may be used to supply an automatic ring again if the called party is busy.

1. TDP: O_Called_Party_Busy
2. DP criterion: Trigger assigned (conditional)
3. Category: Subscribed, Office
4. Trigger assigned to: Subscribed - non-ISDN line, BRI Service Profile, Directory Number and Call Type, BRI/PRI interface; Office - non-ISDN line, ISDN interfaces
5. Trigger Type: oCalledPartyBusy
6. Fault handling: Final Treatment

5.3.6.14 O_No_Answer

The SSF/CCF detects the O_No_Answer trigger when the application timer associated with the O_No_Answer event expires: a time-out associated with the originating portion of the call occurs. For example, this trigger type may be used to provide automatic alternate routing when the timeout occurs.

1. TDP: O_No_Answer
2. DP criterion: Trigger assigned (conditional)
3. Category: Subscribed, Office
4. Trigger assigned to: Subscribed - non-ISDN line, BRI Service Profile, BRI/PRI interface, Directory Number and Call Type; Office - non-ISDN line, ISDN interfaces
5. Trigger Type: oNoAnswer
6. Fault handling: Final Treatment

5.3.6.15 O_Answer

The SSF/CCF detects the O_Answer trigger when answer indication from the terminating BCSM is received.

1. TDP: O_Answer
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group
5. Trigger Type: oAnswer
6. Fault Handling: Call continued.

5.3.6.16 O_Disconnect

The SSF/CCF detects the O_Disconnect trigger when either the call is cleared from the terminating BCSM or the originating facility disconnects.

1. TDP: O_Disconnect
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group
5. Trigger Type: oDisconnect
6. Fault Handling: Final treatment if clearing from the terminating BCSM, continue call processing if the originating facility disconnects.

5.3.6.17 O_Switch_Hook_Flash_Immediate

The O_Switch_Hook_Flash_Immediate trigger is detected in the originating BCSM when the SSF/CCF detects a switch hook flash indication from a non-ISDN line. Note the unconditional nature of the O_Switch_Hook_Flash_Immediate trigger.

1. TDP: O_Mid_Call
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line
5. Trigger type: oSwitchHookFlashImmediate
6. Fault handling: No administrable fault handling (continue call processing)

5.3.6.18 O_Switch_Hook_Flash_Specified_Code

The SSF/CCF detects the O_Switch_Hook_Flash_Specified_Code trigger in the originating BCSM after it receives a switch hook flash indication from a non-ISDN line, applies second dial tone to the non-ISDN line, and collects a feature code (e.g., *XX) or an access code (e.g., 8+) that is designated as the criterion for this trigger at the SSF/CCF. Note the conditional nature of the O_Switch_Hook_Flash_Specified_Code trigger.

1. TDP: O_Mid_Call
2. DP criteria: Trigger assigned (conditional), feature code or access code
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line
5. Trigger type: oSwitchHookFlashSpecifiedCode
6. Fault handling: No administrable fault handling (continue call processing)

5.3.6.19 Termination_Attempt

The SSF/CCF detects the Termination_Attempt trigger when the call directed to the DN/CT reaches the Termination_Attempt DP.

1. TDP: Termination_Attempt
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: Directory Number and Call Type (equipped or unequipped).
5. Trigger Type: terminationAttempt
6. Fault Handling: Final treatment

5.3.6.20 T_Busy

The SSF/CCF detects the T_Busy trigger when it determines that the terminating access is busy (i.e., network determined user-busy). For example, this trigger may be used to forward the call to another number (based on time of day) if the terminating access is busy.

1. TDP: T_Busy
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, Directory Number and Call Type, BRI/PRI interface
5. Trigger Type: tBusy
6. Fault handling: Final Treatment

5.3.6.21 Term_Resource_Available

The SSF/CCF detects the Term_Resource_Available trigger when it determines that the terminating access is not busy (i.e., an idle facility [e.g., B-channel, call appearance, or trunk] could be found). The conditions that result in the detection of terminating resource available depend on the type of terminating access and subscribed services.

1. TDP: Facility_Selected_and_Available
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group, Directory Number and Call Type.
5. Trigger Type: termResourceAvailable
6. Fault Handling: Final treatment

5.3.6.22 T_No_Answer

The SSF/CCF detects the T_No_Answer trigger when the application timer associated with the T_No_Answer event expires: a time-out associated with the terminating portion of the call occurs. For example, this trigger type can be used to automatically re-route the call to a voice mail system.

1. TDP: T_No_Answer
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line, Directory Number and Call Type, BRI/PRI interface, BRI Service Profile
5. Trigger Type: tNoAnswer
6. Fault handling: Final Treatment

5.3.6.23 T_Answer

The SSF/CCF detects the T_Answer trigger when it detects an answer indication from the terminating facility.

1. TDP: T_Answer
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed

4. Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group
5. Trigger Type: tAnswer
6. Fault Handling: Call continued

5.3.6.24 T_Switch_Hook_Flash_Immediate

The T_Switch_Hook_Flash_Immediate trigger is detected in the terminating BCSM when the SSF/CCF detects a switch hook flash indication from a non-ISDN line. Note the unconditional nature of the T_Switch_Hook_Flash_Immediate trigger.

1. TDP: T_Mid_Call
2. DP criterion: Trigger assigned (unconditional)
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line
5. Trigger type: tSwitchHookFlashImmediate
6. Fault handling: No administrable fault handling (continue call processing)

5.3.6.25 T_Switch_Hook_Flash_Specified_Code

The SSF/CCF detects the T_Switch_Hook_Flash_Specified_Code trigger in the terminating BCSM after it receives a switch hook flash indication from a non-ISDN line, applies second dial tone to the non-ISDN line, and collects a feature code (e.g., *XX) or an access code (e.g., 8+) that is designated as the criterion for this trigger at the SSF/CCF. Note the conditional nature of the T_Switch_Hook_Flash_Specified_Code trigger.

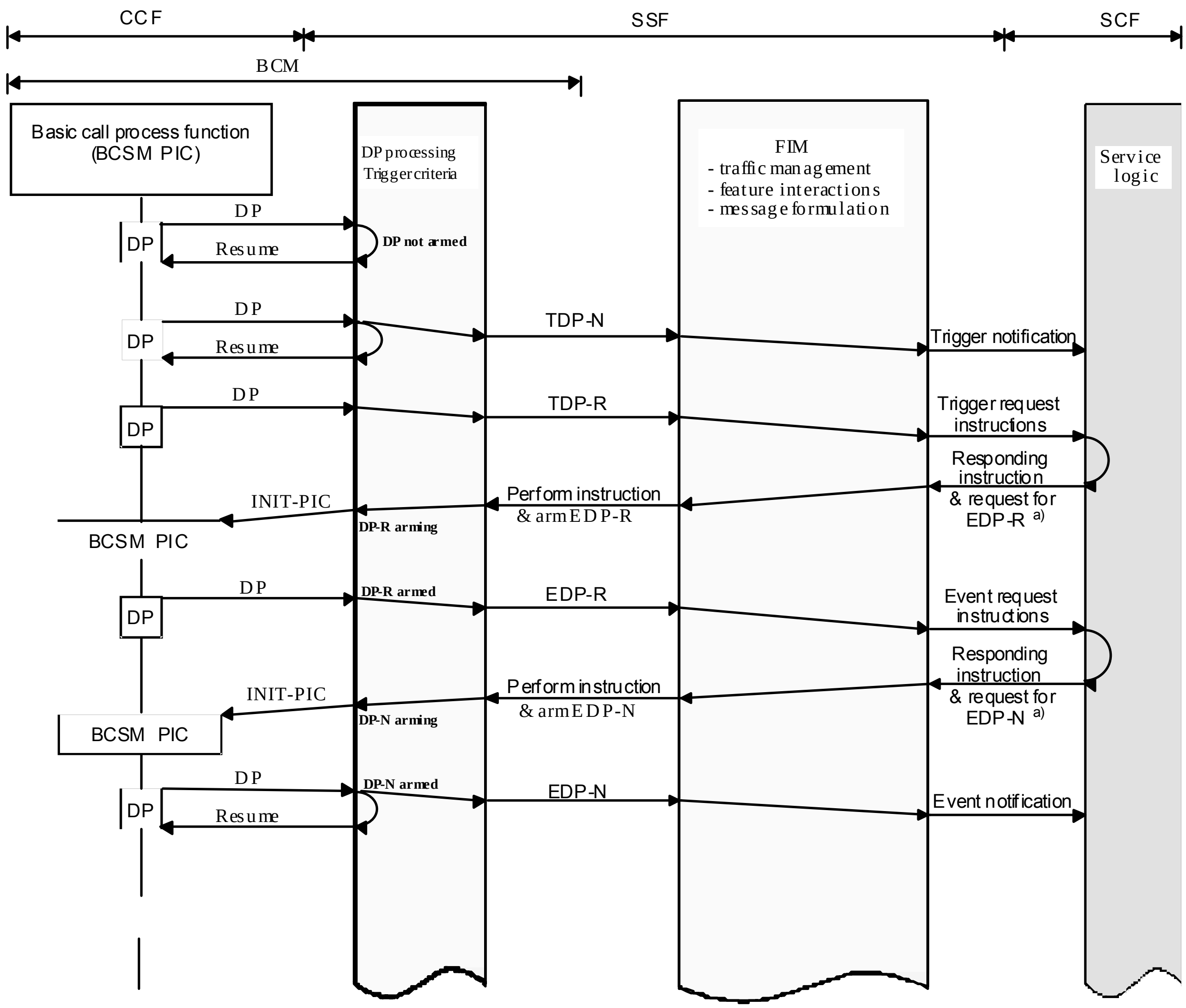
1. TDP: T_Mid_Call
2. DP criteria: Trigger assigned (conditional), feature code or access code
3. Category: Subscribed
4. Trigger assigned to: non-ISDN line
5. Trigger type: tSwitchHookFlashSpecifiedCode
6. Fault handling: No administrable fault handling (continue call processing)

5.3.7 DP processing

DP processing involves:

- traffic management actions ;
- determining if DP criteria are met (see 5.3.5 and this clause);
- handling service logic instance interactions when invoking new instances of IN and non-IN service logic;
- and formulating operations to send to one or more SCFs.

DP processing for each DP type is illustrated in Figure 8.



a) In this example, the responding instruction and request for EDP are shown together. These are independent information flows and may not be sent together in all cases.

DP	Detection point
TDP	Trigger detection point
EDP	Event detection point
R/N	Request/notification
PIC	Point in call

T1136250-91/d07

Figure 8 - DP processing for each DP type

Since a DP may be armed as a TDP and/or an EDP for the same call, the BCSM should apply the following set of rules during DP criteria processing to ensure single point of control:

Rule 1: At any DP, a specific trigger condition can only trigger one service logic program instance (SLPI) at a time.

Rule 2: At any DP, processing of notifications - EDP-N and TDP-N - has higher priority than processing of requests - EDP-R and TDP-R. If several notifications exist, EDP-R and TDP-R are processed when all notifications have been processed.

Rule 3: If a DP is both armed as EDP and TDP, then the EDP processing has higher priority than the TDP processing since the EDP has been armed in an already existing SSF-SCF relationship.

Rule 4: If a DP is both armed as EDP-R and TDP-R, The EDP-R is first processed and, if the control relationship is terminated as a result of the EDP-R processing, processing of the TDP-R is allowed.

The rules are listed in descending priority order. They are illustrated with the diagram in Figure 9.

A control relationship persists as long as there is at least one EDP-R armed for at least one call segment of a call segment association. A control relationship terminates if there are no more EDPs armed or the call clears. During a control relationship, EDPs may be dynamically disarmed by the SCF, or are disarmed by the SSF as they are encountered and reported to the SCF, or when the call clears.

Single point of control ensures that only one service logic exists within the control relationship.

Single point of control is only guaranteed within a Call Segment Association.

A control relationship changes to a monitor relationship if there are no more EDP-Rs armed and at least one EDP-N armed. A monitor relationship terminates if there are no more EDP-Ns armed or the call clears. During a monitor relationship, EDP-Ns are disarmed by the SSF as they are encountered and reported to the SCF, or when the call clears.

TDP-N criteria may be processed whether or not there is an existing control relationship for the same portion of the call, since a TDP-N does not open a control relationship. This procedure has no effect on the existing control relationship.

NOTE - It is possible to cause infinite retriggering during the processing of a call. Such an example is as follows:

1) User dials 555-1111
2) Analyzed_Information DP is armed as TDP-R, and the criterion is 555-1111
3) SCF returns a Destination Routing Address of 555-1111
4) The call will resume at the Analyze_Information PIC and then trigger at the Analyzed_Information DP again. This effectively produces an infinite loop.

It should be noted that this loop can occur any time the response from a DP returns the call to a previous point in the BCSM. Two possible solutions to the problem (others may be possible) that have been identified are identified below, but none of these are mandatory.

A) Letting the SCF prevent multiple retriggering.
This solution means that the SCF contains enough logic that it will never return the same *Called Party Number* as was received, if the trigger criterion were the *Called Party Number*. If the trigger criteria were something else, such as Shared Interoffice Trunk, then returning the same *Called Party Number* is not an issue.

B) Letting the SSF prevent multiple retriggering
This solution means the SSF will count the number of times a call is triggered to the SCF, on a per half call basis. If the count exceeds a set limit, the SSF could take appropriate action (e.g., the call could be taken down or another suitable action).

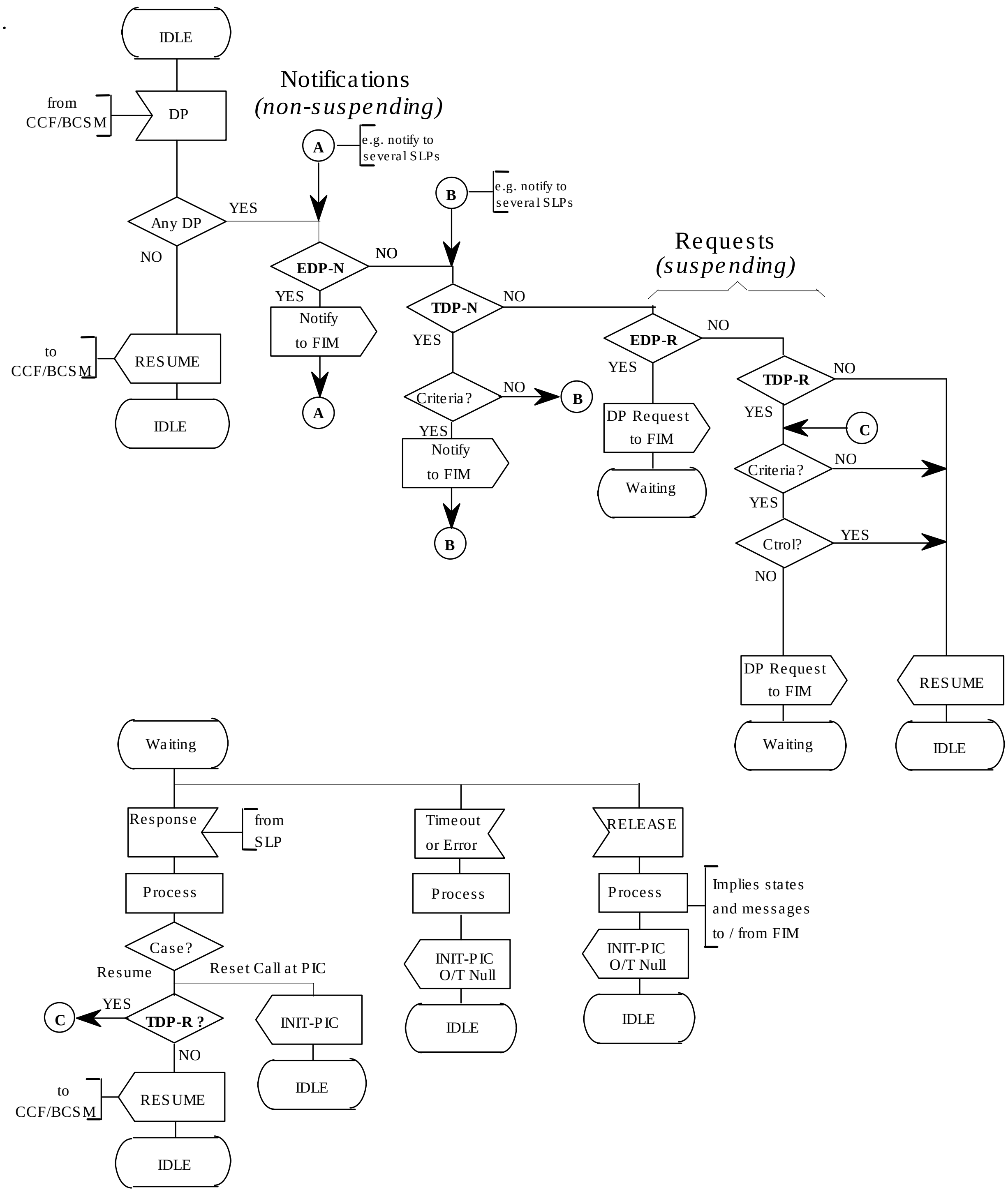
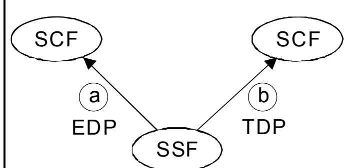
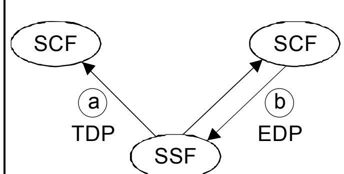
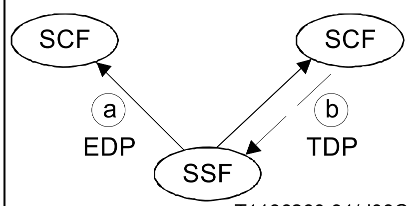


Figure 9 - Detection point processing

As a consequence of these rules, the BCSM should support a number of TDP/EDP processing combinations to ensure single point of control. These combinations are identified in Table 7, along with the error combinations that should not occur:

Table 7 - TDP/EDP processing combinations

Scenario	TDP type	EDP type	Existing relationship	Processing
1	Not armed	Not armed	Don't care	Continue
2	TDP-R	Not armed	No	Initiating DP request
3.a	TDP-R	Not armed	Control	Continue (ignore TDP)
3.b	TDP-R	Not armed	Monitor	Initiating DP request
4	TDP-N	Not armed	Don't care	One-way DP notification Initiate Monitor relationship
5.a	Not armed	EDP-R	Control	Subsequent DP request, if at least one armed EDP remains, or terminating DP request, if last armed EDP
5.b	Not armed	EDP-R	Monitor	Error case – Continue (ignore EDP)
6	Not armed	EDP-N	Control or Monitor	Subsequent DP notification, if at least one armed EDP remains, or terminating DP notification, if last armed EDP
7	Not armed	EDP-R/N	No	Error case – Continue (ignore EDP)
8	TDP-N	EDP-N	Control or monitor	Process a and b: a) For EDP, Process as scenario 6 b) For TDP, one-way DP notification, Process as scenario 4 
9	TDP-N	EDP-R/N	No	Error case – Ignore EDP and process TDP as scenario 4
10.a	TDP-N	EDP-R	Control	Process a and b: a) For TDP, One-way DP notification, Process as scenario 4 b) For EDP, Subsequent DP request, Process as scenario 5a 
10.b	TDP-N	EDP-R	Monitor	Error case – Ignore EDP and process TDP as scenario 4

Scenario	TDP type	EDP type	Existing relationship	Processing
11.a	TDP-R	EDP-N	Control	Process a and b: a) For EDP, Subsequent DP notification, Process as scenario 6 b) Ignore TDP
11.b	TDP-R	EDP-N	Monitor	Process a and b: a) For EDP, Process as scenario 6 b) For TDP, Initiating DP request, Process as scenario 3b 
12	TDP-R	EDP-R/N	No	Error case – Ignore EDP and process TDP as scenario 2
13.a	TDP-R	EDP-R	Control	Process EDP as scenario 5a. If this EDP was the last of the previously established control relationship, process the TDP afterwards. Otherwise, the TDP is ignored.
13.b	TDP-R	EDP-R	Monitor	Error case – Ignore EDP and process TDP as scenario 3.b

5.4 IN-switching manager (IN-SM)

The IN-SM centers on the IN-switching state model (IN-SSM) that provides a description of SSF/CCF IN call/connection processing in terms of IN call/connection states. Object-oriented techniques are used to describe the IN-SSM.

The IN-SM subjects described in the following clauses include the IN-SSM, IN-SSM events that can be reported to active IN service logic instances, and SSF resource control. A high-level description of these subjects is provided.

5.4.1 IN-switching state model (IN-SSM)

The IN-SSM provides an object-oriented finite state machine description of SSF/CCF IN call/connection processing in terms of IN call/connection states. It provides a framework for describing the scope of view and control of SSF/CCF activities offered to an SCF. The extent to which the IN-SSM is visible to the SCF is defined by the information flows identified for T1 IN between the SSF/CCF and SCF. Though this framework is consistent with the scope of T1 IN, not all of the capabilities implied by the IN-SSM are supported by the operations and parameters defined in clause 7.

IN call/connection states can be described in terms of the IN-SSM, which defines the set of SSF/CCF objects visible to the SCF. Each IN-SSM instance provides the SCF with a limited aperture of visibility and influence into SSF/CCF IN call/connection processing. This aperture of visibility and influence is defined by the objects that constitute the IN-SSM. These objects are abstractions of SSF/CCF resources accessible to the SCF.

There can be various types of IN-SSMs, each type defined by the objects that constitute it. For example, a “connection control” IN-SSM would contain objects that are abstractions of switching and transmission resources. This clause focuses on such a connection control IN-SSM, though it is recognized that other types of IN-SSMs may exist for accessing other types of resources.

There can also be various subtypes of a particular IN-SSM type, each defined by a subset of, or restriction on the use of, the total set of objects in the IN-SSM type. It is anticipated that IN-SSM subtypes will be identified to align with specific IN capability sets as they are defined.

A connection control IN-SSM instance is created when an IN service logic instance is invoked that requires IN connection control. It is either created as a result of encountering a TDP in a BCSM that satisfies DP criteria, or is initiated by the SCF independent of encountering TDPs. A connection control IN-SSM instance is destroyed when the SCF informs the SSF that the IN service logic instance is completed or the IN-SSM should be destroyed. The SSF can also initiate IN-SSM destruction (e.g., during error or abnormal conditions).

Figure 10 provides an example of a connection control IN-SSM instance. It illustrates two classes of objects that have been identified: legs and connection points. A leg is a representation of a communication path towards an addressable network entity, as viewed from the IN-SSM. A connection point is a representation of the interconnection of legs, as viewed from the IN-SSM that allows information to flow between legs. It should be noted that the fundamental processes that establish communication paths, and maintain connections between them, are the basic call processes modeled by one or more BCSMs. As such, the connection control IN-SSM objects reflect both connectivity information (e.g., the relation of legs and connection points to each other) and call processing information (e.g., BCSM events and basic call-related information), which can be used by an instance of IN service logic to influence the connectivity and call processing aspects of a call.

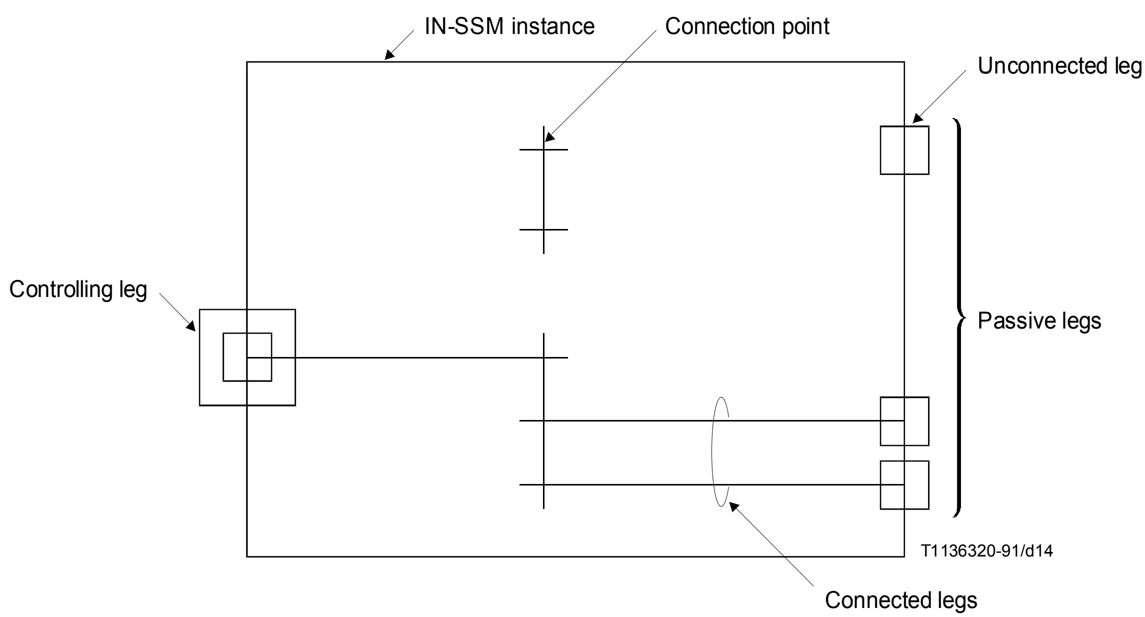


Figure 10 - Connection control IN-SSM instance

The attributes of these objects and their relation to each other describe the state of the connections, and supporting basic call processes, represented by the IN-SSM. The SCF can invoke SSF functions that manipulate these objects (e.g., changing their attributes or their relationship to each other, thereby changing the state of the connections and supporting basic call processes). This state information is provided to the SCF via information flows and information elements (e.g., EDP-Request information flows and related information elements). The allowable state changes for T1 IN are reflected in the semantic description of SCF-SSF information flows related to basic call processing.

Objects in an IN-SSM are controlled within the context of an SCF-SSF interaction as defined by the IN-SSM type. As such, they are considered local to the IN-SSM. However, manipulating an object can have significance outside the IN SSM and the SSF. In particular, leg manipulation has significance beyond the boundaries of the IN-SSM, even though it can only be controlled within the context of the IN-SSM. This is because a leg represents a path toward some addressable entity that may be supported by switching and transmission resources beyond the immediate control of the SSF/CCF. Thus, the relation between leg manipulation and signaling associated with switching and transmission resources should be identified as part of the semantic description of SCF-SSF information flows related to leg manipulation.

Other object classes related to the connection control IN-SSM are abstractions of specialized resources such as tones and announcements. These objects will not be explicitly shown in a connection control IN-SSM for T1 IN (though they may appear in the context of other IN-SSM types, such as “resource management” IN-SSMs). However, they may be implicitly used within a connection control IN-SSM via SSF functions that manipulate connection control IN-SSM objects (e.g., functions to send/receive information to/from users via legs). In addition, their use may be reflected in a connection control IN-SSM as an appearance of a leg representing a path to some external entity that provides specialized resources (such as may be supported by an SRF). Other object classes are not explicitly modeled for T1 IN, though they are implied by other information flows/information elements defined for T1 IN.

The characteristics of SSF/CCF call processing represented by connection control IN-SSM objects for T1 IN are described below. These characteristics imply the attributes and functions related to IN-SSM objects, to be reflected in the call processing information flows/information elements defined for T1 IN.

A. The T1 IN connection control IN-SSM provides the SCF with an abstract view of an isolated portion of a call managed by a functionally separate portion of the SSF/CCF. This isolated portion of a call is referred to as a “half-call” or call segment (see Figure 11).

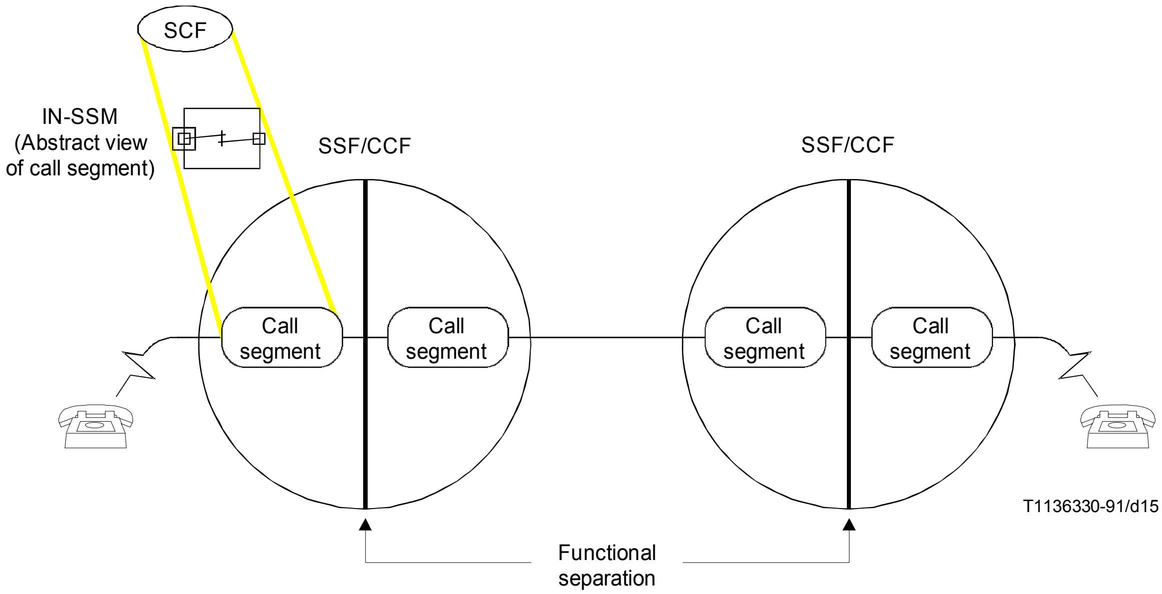


Figure 11 - Call segments in two-party inter-SSF/CCF call

B. This term is used to refer to the physical resources (e.g., connectivity and transmission resources represented by legs and connection points) and to the processes (e.g., basic call processes as modeled by BCSMs) that are involved in the isolated portion of a call (per Figure 3 and Figure 4).

The SCF does not have direct access to a call segment, but rather has access to the abstract representation of the call segment provided by the T1 IN connection control IN-SSM. For T1 IN, access via an T1 IN connection control IN-SSM is limited to a single two-party or multi-party call segment, or to a pair of associated call segments (see Figure 12). (With respect to the term “multi-party call segment”, note that only single-ended, single-point-of-control multi-party call segments are within the scope of T1 IN.) A pair of associated call segments are two call segments that can be related together by the SSF/CCF and manipulated as a pair (e.g., to merge them together into a single call segment). For T1 IN, two call segments can only be associated if both call segments are for the same end user. For example, the SSF/CCF can associate two call segments if the end user is involved in an existing call and wants to originate an additional call, or if the end user is involved in an existing call and there is a new call directed to that end user. This latter example is shown in Figure 12.

C. An T1 IN connection control IN-SSM provides an SCF with an abstract view of a single two-party or multi-party call segment, or of a pair of associated call segments. The connection control IN-SSM represents the properties of a call segment or pair of associated call segments of interest to the SCF (e.g., the connectivity and call processing aspects) and describes these properties in terms of objects (i.e., virtual resources) that can be manipulated by the SCF. For connection control, these objects include legs and connection points.

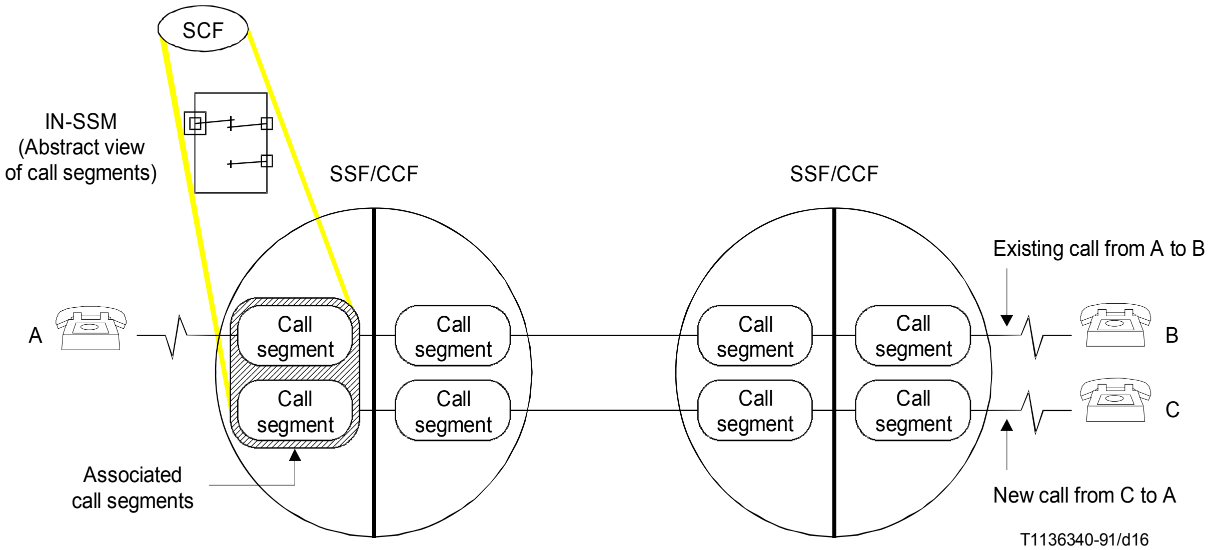


Figure 12 - Associated call segments

- A leg can be designated as a controlling leg or as a passive leg. For T1 IN, the controlling leg is the leg that represents the access interface (e.g., the incoming line or trunk in an originating call segment, or the outgoing line or trunk in a terminating call Segment). It is the leg for which IN service logic instances are invoked, either as a result of end user signaling (e.g., a mid-call event) or on behalf of an end user. There is no more than one controlling leg in a connection control IN-SSM. Transfer of control from an end user supported by a controlling leg to an end user supported by a passive leg is not feasible for T1 IN.
- For T1 IN, controlling legs represent line or trunk interfaces. There may be a limitation on how these two types of controlling legs may be manipulated by the SCF for T1 IN.
- Legs are uniquely identifiable in an T1 IN connection control IN-SSM.
- It should be possible to: influence the flow of basic call processing associated with a leg (e.g., generate a signaling event and continue basic call processing as appropriate for that event); add a passive leg to a T1 IN connection control IN-SSM by originating a call or terminating a call; to drop legs (one or more) by clearing calls; to make or break connections between legs (e.g., join or split); and to move legs from one connection point to another within the same T1 IN connection control IN-SSM (e.g., split a leg from one connection point, then join that leg to another).
- A connection point represents a joint function between two legs, a conference function among three or more legs, or an information distribution function between two or more legs that specifies the directionality of information flow through the connection point (e.g., the connection point could receive information from multiple legs and distribute it to another leg). For T1 IN, it interconnects legs supported by equivalent bearer services, and supports interworking between circuit mode/speech and circuit mode/3.1 kHz audio bearer services.
- There can be up to two connection points in an T1 IN connection control IN-SSM, one per call segment that is represented by the IN-SSM. There can only be two connection points if call processing for one of the call segments has progressed beyond call setup. Further, only one of the two connection points in an T1 IN connection control IN-SSM can interconnect more than two legs. The other connection point can only interconnect two

legs. In an T1 IN connection control IN-SSM, it should be possible to merge two connection points into a single connection point, thereby merging the corresponding call segments. Finally, it should be possible to release a connection point and all of its legs all at once, thereby clearing the corresponding call segment.

The call segment concept can be used to describe how the definitions of "*single-ended service feature*" and "*single point of control*" apply to the distributed functional plane.

A *single-ended service feature* is described in terms of:

- the scope of control of the service logic instance that realizes the service feature, with respect to the call; and
- the interaction of the service logic instance with respect to other single-ended service logic instances on the same call.

The scope of control of a single-ended service logic instance is restricted to the isolated "half-call(s)" in an SSF/CCF (i.e., the call segments) accessible to the SCF via a control relationship. This is illustrated in Figure 13 for a two-party call, which shows the BCSMs related to each call segment.

This may also be extended in T1 IN for a pair of associated "half-calls", or a multi-party "half-call". These scenarios are illustrated in Figure 14 and Figure 15.

All of these scenarios are based on the assumption that "half-calls" can be isolated from their complementary "half-calls" by the functional separation between an originating BCSM and its complementary terminating BCSM.

A single-ended service logic instance can only directly influence the processing of the isolated "half-call" (or associated "half-calls") in the SSF/CCF. The other "half-calls" can only be indirectly influenced via information propagating from one "half-call" to another (i.e., between originating and terminating BCSMs, or between BCSMs in different SSF/CCFs). As such, multiple single-ended service logic instances (one per "half-call") may be simultaneously active on a single call, each isolated from the other by the communication between "half-calls". The communication between originating and terminating BCSMs in the same SSF/CCF is described in 5.3.2 (BCSM). The communication between BCSMs in different SSF/CCFs is assumed to be the same as existing signaling between exchanges.

Single point of control, as it applies to the distributed functional plane is as follows:

- an isolated "half-call" in the SSF/CCF can only be influenced by one SCF at a time;
- while one SCF is influencing an isolated "half-call" in the SSF/CCF, it may be possible to;
- send TDP-N/EDP-N information flows from the SSF/CCF to the same SCF or different SCFs;
- end the control relationship between the controlling SCF and the SSF/CCF, or change the control relationship to a monitor relationship, then initiate a control relationship between the SSF/CCF and a different SCF.

5.4.2 IN-SSM EDPs

Certain IN-SSM events can be reported to active IN service logic instances that have already been invoked. These events are referred to as IN-SSM EDPs. For example, events such as the successful completion or failure of a particular IN-SSM function may need to be reported. Detection of IN-SSM EDPs does not lead to the invocation of additional IN service logic instances. IN-SSM EDPs are handled implicitly for T1 IN for those information flows from the SCF that require confirmation by the SSF/CCF.

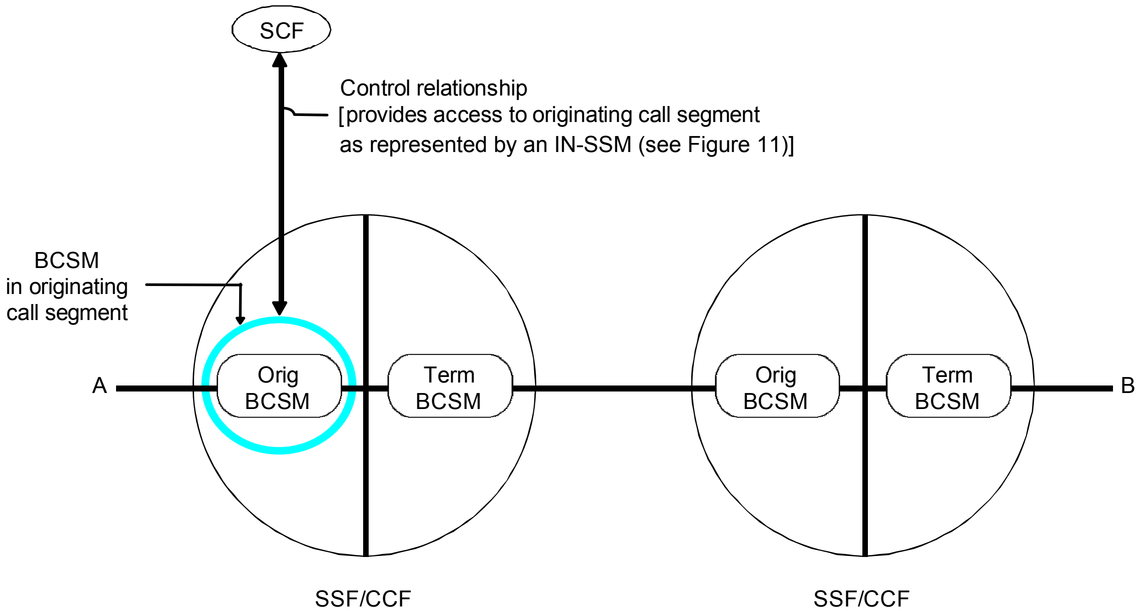


Figure 13 - Single-ended control of a two-party call

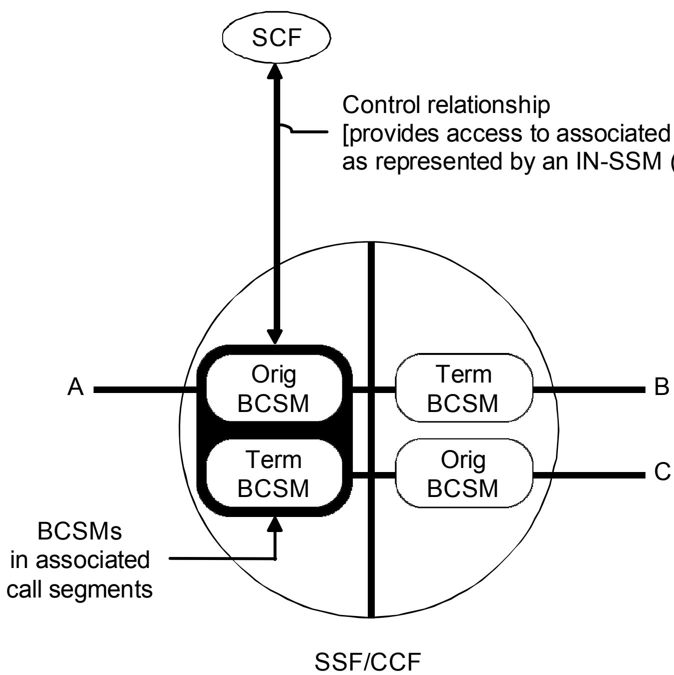


Figure 14 - Single-ended control of associated calls

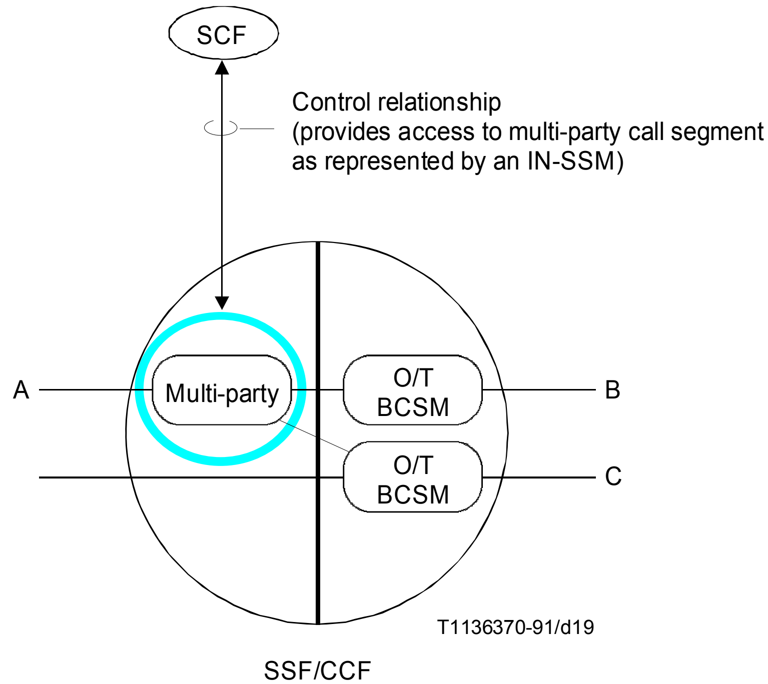


Figure 15 - Single-ended control of a multi-party call

5.4.3 SSF resource control

Local and remote specialized resources needed to perform IN call/service processing are accessible to the SSF/CCF. The treatment of specialized resources with respect to the connection control IN-SSM was described above. Objects that explicitly represent specialized resources are not explicitly modeled for T1 IN, though they are implied by other information flows/information elements defined for T1 IN.

5.5 Relationship of SSF/CCF to SCF

This clause only addresses call associated relationships as supported by an T1 IN connection control IN-SSM.

- An SSF/CCF can have call associated relationships with multiple SCFs, and an SCF can have call associated relationships with multiple SSF/CCFs. Each relationship is treated as a one-to-one relationship.
- When the SSF/CCF initiates a relationship, it reports the state of the IN-SSM in which the TDP was detected. The state information that is included in the information flows between the SSF/CCF and the SCF is defined by the information elements included in the information flows, based on the analysis of detailed DFP modeling.
- Once a control relationship is established between the SSF/CCF and the SCF, the SCF can request the SSF/CCF to monitor for and report subsequent events (i.e., arm EDPs), as well as to stop monitoring (i.e., disarm EDPs).

5.6 Specialized resource function (SRF) model

5.6.1 General

A model of the SRF is shown in Figure 16. The purpose of this model is to provide a framework for specialized resource functionality subjects with respect to the SRF.

SRF provides various specialized resources as shown in 5.6.4. The SRF is managed to place resources in or out of service, e.g., for provisioning, administration and maintenance purpose.

For call/service processing, the SRF has a logical relationship with the SSF/CCF and the SCF. The SCF controls the connection between the SSF/CCF and the SRF, and sends instructions to the SRF.

As part of the process of formulating a response to the SSF, the SCF may need to enter into a dialogue with a calling or a called party. This could, for example, take the form of a prompt and collect digits sequence.

The SCF in T1 IN will instruct the SRF to start a dialogue with a user after setting up a path between the SSF/CCF and the SRF. The dialogue between the SRF and the user allows the SRF to play an announcement and if appropriate, collect digits. If digits have been collected, the SRF will pass the digit information to the SCF.

A user, being prompted from the SRF, inputs MF tones, for example, to the SRF so that the collected digits can be reported to the SCF. When the service logic in the SCF does not need the resources anymore, the SCF can either request the SRF to release the connection to the SSF/CCF or request the SSF/CCF to release the connection with the SRF and the resource in the SRF will be released.

5.6.2 SRF components

To provide the functionality defined in the previous clause, the SRF includes the following functions as illustrated in Figure 16.

- Functional entity access manager (FEAM)
- The FEAM provides the functionality necessary for the SRF to exchange information with other functional entities via messages as follows:
    - provide reliable message transfer;
    - ensure sequential message delivery;
    - allow message request/response pairs to be correlated;
    - allow multiple messages to be associated with each other;
    - comply with OSI structures and principles.

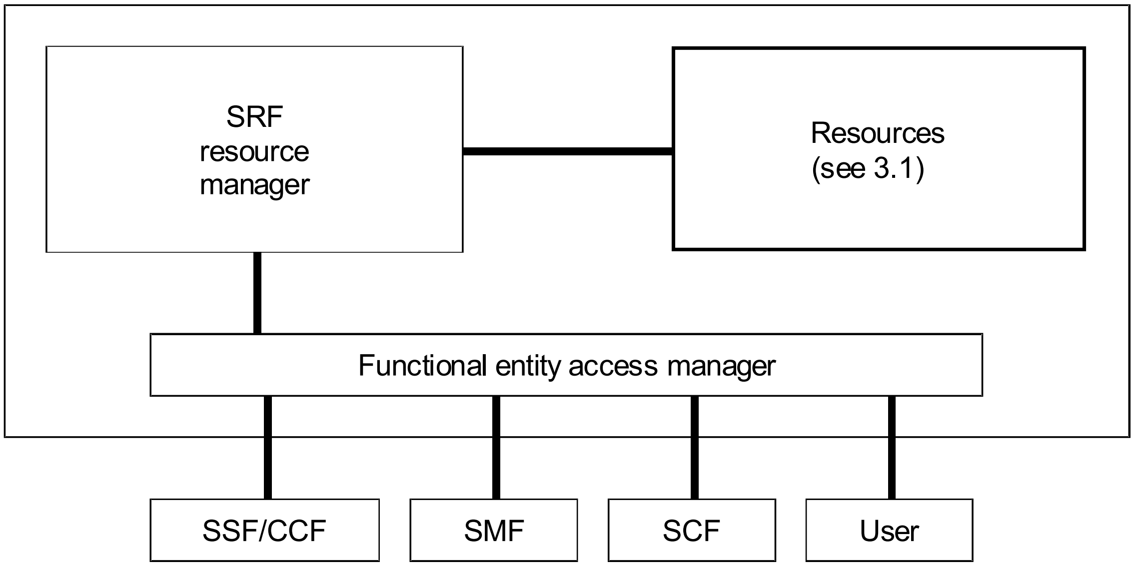


Figure 16 - SRF model

- SRF resource manager (RM)

  The SRF RM provides the necessary functionality to manage resources contained in the SRF. This contains the capabilities to hunt for a resource, to manage resource status, e.g., busy/idle/block, and to control resource actions.

- Resources

  The SRF contains various resources, which are listed in 5.6.4.

5.6.3 SRF and other entity relationships

The SRF has a relationship with the SSF/CCF, SCF, user and SMF as follows:

- SSF/CCF

  The SRF has a relationship with the SSF/CCF for connection control to specialized resources.

  The SRF may contain functionality similar to the CCF to manage bearer connections to specialized resources, but no call model is specified.

- SCF

  The SCF sends connection control information to the SSF/CCF. In T1 IN, this relationship is supported by the interface protocol defined in clause 7.

  The connection between an SRF and an SSF/CCF is set up by the SSF/CCF according to the control information received from the SCF. Then, the SCF sends instructions either through the SSF relay method or the direct SCF-SRF method to the SRF for resource manipulation.

- User

The SRF has an information exchange relationship with the user using, for example, a voice channel, ISDN bearer channel, and SS7 trunk connection. The requirements on this user channel are not affected by connection to the SRF.

- SMF

  The SRF provides the SMF with management information and actions requested.

5.6.4 Objects of SRF management

Examples of specialized resources managed by the SRF are as follows:

- DTMF receiver.
- tone generator.
- announcements.
- message sender/receiver.
- synthesized voice/speech recognition devices with interactive prompting facilities.
- text to speech synthesis.
- protocol converters.
- audio conference bridge.
- information distribution bridge.

The following four objects are supported in T1 IN and defined as follows:

- DTMF receiver

  This resource receives dual tone multi-frequency (DTMF) from a linked resource, and recognizes it as a standardized signal input.

- tone generator/announcements

  This resource provides in-channel information to the specified virtual resource.

- message sender/receiver

  This resource sends or receives messages, such as electronic messages, voice messages, etc., to/from users.

- synthesized voice/speech recognition device with interactive prompting facilities

  This resource receives in-channel speech information from a linked virtual resource, and recognizes it as a standardized signal input. When the information is input from a user, it is recognized by this resource and this resource converts it to IN perceivable signals. When this resource receives an instruction to send a voice message with source-information, it is converted to a voice message. Usually, such action is performed with interactive prompting.

5.7 Service control function (SCF) model

5.7.1 General

A model of the SCF is shown in Figure 17. The purpose of this model is to provide a framework for service logic processing subjects with respects to the SCF.

The prime function of the service control function (SCF) is execution of service logic provided in the form of service logic processing programs (SLPs) and, accordingly, it includes the SLP execution supporting functions, such as service logic selection/interaction management, functional entity access management, and SLP provisioning management.

5.7.2 SCF components

5.7.2.1 General

To realize the above defined functionality, the SCF model is shown in Figure 17. It is noted that this shows a conceptual model of SCF and is not intended to imply an actual implementation of the SCF.

The SCF platform provides a service logic execution environment (SLEE) on which a service logic processing program (SLP) runs to provide pertinent service processing. An SLP is a service application program invoked by the SLEE and is used to realize service processing under the control of the SLEE. The simultaneous invocation and execution of multiple SLPs are also managed by the SLEE.

Each of the entities shown in Figure 17 is described in the successive clauses.

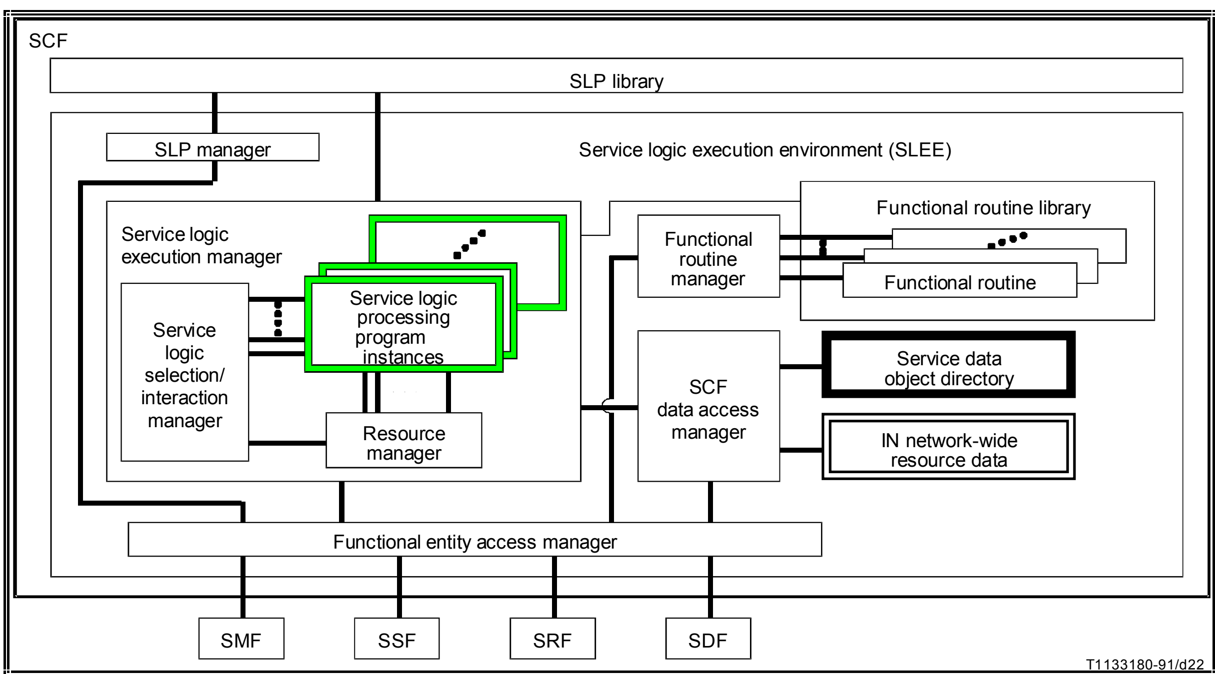


Figure 17 - SCF model

5.7.2.2 Service logic execution manager (SLEM)

5.7.2.2.1 General

The SLEM is the functionality that handles and controls the total service logic execution action. The SLEM contains service logic processing program instances (SLPIs), service logic selection/interaction manager, and resource manager. It also interacts with SCF data access manager and functional entity access manager to support SLPI execution. In addition to these aspects, the SLEM needs functionality to:

- execute SLPIs and maintain transient data associated with SLPIs (i.e., information that only persists during the lifetime of the SLPI, such as SLPI state information);
- execute functional routines in support of SLPI execution;
- manage SLPI access to SCF and SDF data via the SCF data access manager (see 5.7.2.3);
- manage the exchange of information between SLPIs and entities in other functional entities via the functional entity access manager (see 5.7.2.5).

5.7.2.2.2 Service logic selection/interaction manager (SLSIM)

The SLSIM is the entity that selects an SLP for execution and controls the simultaneous execution and/or execution order of multiple SLPs in the same SCF.

As part of the functionality, the SLSIM provides a means to manage service interactions by managing interactions among multiple SLPIs in the same SCF that are simultaneously active on a single call.

SLP selection is performed via the SLSIM in response to:

- an external event from another functional entity;
- the occurrence of internally recognized conditions (e.g., time of day or other internal events);
- the execution of a functional routine via an SLPI that requests the execution of another SLP.

In addition, the SLSIM should invoke the execution of the selected SLP and provide for mutual exclusion and precedence during this SLP selection and invocation:

- mutual exclusion prevents the invocation of an SLP whose execution would be incompatible with currently executing SLPI;
- precedence provides a scheme to select a particular SLP from a set of SLPs that meet the same selection criteria.

5.7.2.2.3 Service logic processing program instance (SLPI)

A service logic processing program (SLP) is a service application program invoked by the SLEE and used to realize service processing. It contains logical constructs that, when executed, control the flow of service execution, and statements that, when executed, invoke functional routines in the SCF to access network resources and data needed for service execution. When an SLP is selected and invoked, it is referred to as a service logic processing program instance (SLPI). In contrast to an SLP, a corresponding SLPI is a dynamic entity that actively controls the flow of service execution and invokes SCF functional routines. Functional routines are the functionality in the SCF that can be invoked by SLPIs to cause a sequence of functional entity actions to be performed in the network in support of service execution.

5.7.2.2.4 Resource manager

The resource manager provides the functionality to control the allocation of local SCF resources and provides access to network resources in support of SLPI execution. The resource manager contains functionality to:

- identify and locate local SCF resources;
- identify and locate network resources via the SCF data access manager and IN network-wide resource data (see 5.7.2.3.3);
- identify one or more local SCF resources requested by a particular SLPI;
- release one or more local SCF resources no longer needed by a particular SLPI;
- interact with other functional entities via the functional entity access manager to provide for the reservation and release of network resources to be used by SLPIs.

It is noted that SRF selection is not always performed by the SLEM resource manager, in some cases, selection is performed by an SSF.

5.7.2.3 SCF data access manager

5.7.2.3.1 General

The SCF data access manager provides the functionality needed to provide for the storage, management, and access of shared and persistent information in the SCF (i.e., information that persists beyond the lifetime of a SLPI). The SCF data access manager also provides the functionality needed to access remote information in SDFs. The SCF data access manager interacts with the SLEM to provide these functionalities to SLPIs.

Figure 17 identifies two structures that contain SCF data. These include:

- the service data object directory;
- the IN network-wide resource data.

These are described in the following clauses.

5.7.2.3.2 Service data object directory

Figure 17 identifies a service data object directory. This directory provides a means to address the appropriate SCF for access to a specific data object.

The SLEM interacts with the SCF data access manager to access service data objects in SDFs. The SCF data access manager uses the service data object directory to locate service data objects in the network in a manner transparent to the SLEM (and its SLPI). As such, the SLEM (and its SLPIs) has a global and uniform view of service data objects in the network.

5.7.2.3.3 IN network-wide resource data

This is a structure in which information resides about the location and capabilities of resources in the network accessible to SLPIs. It provides a means to address the appropriate functional entity (e.g., SRF) for access to specific resources with the appropriate capabilities.

The SLEM resource manager interacts with the SCF data access manager to access network resource data. The SLEM resource manager provides SLPIs with access to network resources in a manner transparent to SLPIs. As such, SLPIs have a global and uniform view of resources in the network.

5.7.2.4 Functional routine manager

Functional routine manager will be used for reception and distribution of functional routines to functional routine library via functional entity access manager. This entity also manages the addition, deletion and suspension of a particular functional routine.

Functional routine library is an entity where the actual functional routines are residing.

5.7.2.5 Functional entity access manager (FEAM)

The functional entity access manager provides the functionality needed by the SLEM to exchange information with other functional entities via messages. This message-handling functionality should:

- be transparent to SLPIs;
- provide reliable message transfer;
- ensure sequential message delivery;
- allow message request/response pairs to be correlated;
- allow multiple messages to be associated with each other;
- comply with OSI structures and principles.

5.7.2.6 SLP manager

The SLP manager manages the reception and distribution function of SLPs from other entities. The SLP manager, therefore, interworks with functional entity access manager (FEAM). This entity also manages addition, deletion and suspension of a particular SLP.

5.7.3 Functional routine categories

The following categories of functional routines are proposed as framework for describing the SCF functionality accessible to SLPIs:

- SLPI management functional routines:
  - functional routines to facilitate SLPI initialization and termination;
  - functional routines to invoke other SLPs.
- SLPI communication functional routines:
  - functional routines to support communication between SLPIs.
- Timer management functional routines:
  - functional routines to retrieve the current time and date;
  - functional routines to manage asynchronous timers in the SCF;
  - functional routines to block the invocation of an SLP for a certain defined period.
- Data management interface functional routines:
  - functional routines to access and manipulate SCF data (i.e., service data object directory and IN network-wide resource data) and network data (i.e., in an SDF) globally and uniformly via the SCF data access manager.

- Asynchronous event handling functional routines:
  - functional routines to perform appropriate functions in response to asynchronous events (e.g., events reported by other functional entities, SLPI execution error events, and internal SCF events);
  - functional routines to facilitate termination of a service execution and initialization of related resources.
- Connection management functional routines:
  - functional routines to manipulate legs and connection points via interaction with the IN-feature manager in the SSF.
- Specialized resource management functional routines
  - functional routines to access and use specialized network resources globally and uniformly via the SLEM resource manager (interacting with the SRF).
- OAM functional routines:
  - functional routines to respond to request for OAM activities and gather OAM-related information (e.g., data collection, traffic management, error handling, charging).

5.8 Service data function (SDF) model

5.8.1 General

A model of the SDF is shown in Figure 18. The purpose of this model is to provide a framework for service data functionality subjects with respect to the SDF.

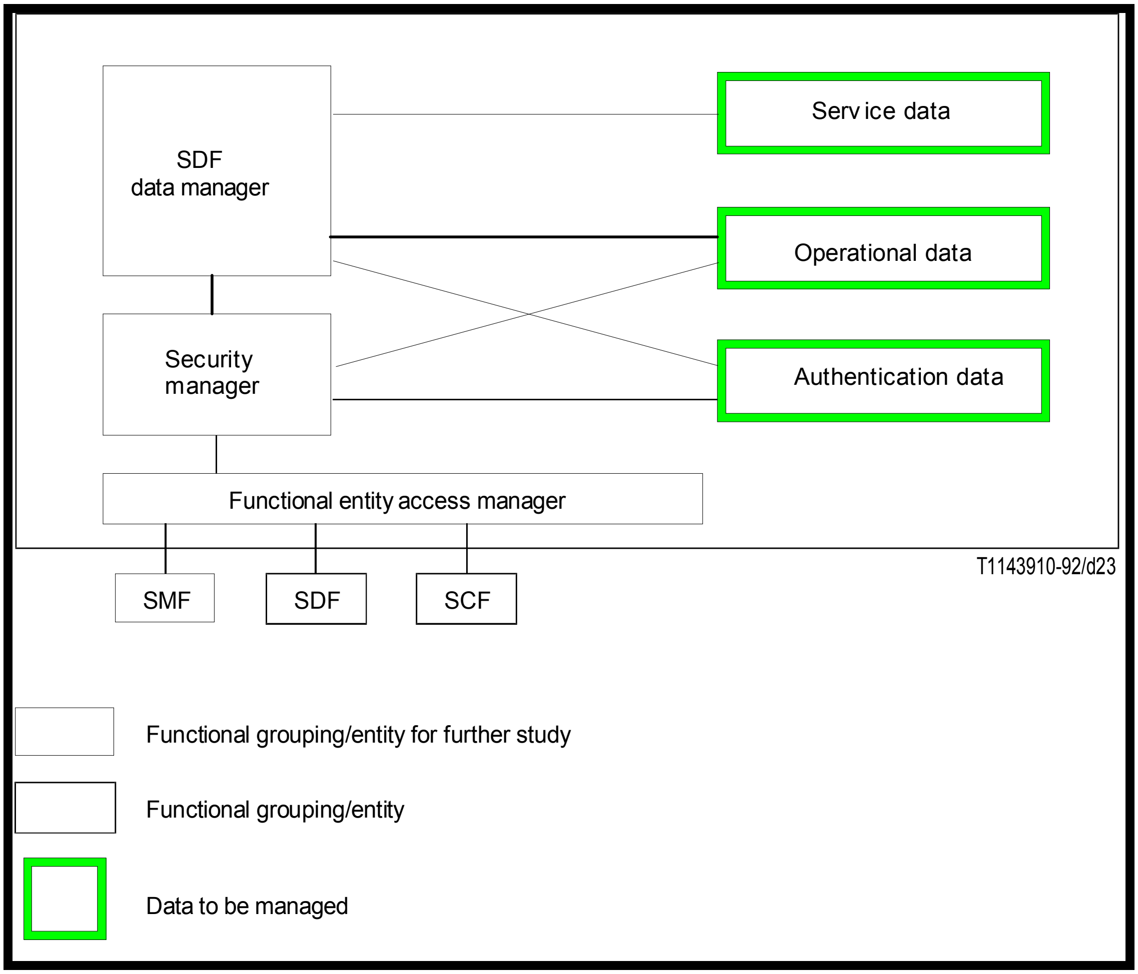


Figure 18 - SDF model

5.8.2 describes the detailed SDF architecture, and 5.8.3 clarifies and classifies data types that are handled by the SDF.

The SDF contains and manages the data that are related to service logic processing programs (SLPs) and accessed in the execution of the SLP instances (SLPIs). Therefore, data such as SLP selection data and SCF directory, which are accessed before the execution of an SLPI, are not included in the SDF handling data.

5.8.2 SDF components

5.8.2.1 General

To realize the above defined functionality, the recommended SDF model is shown in Figure 18. Each of the functional entities shown in Figure 18 is described in the successive clauses. This is not meant to imply any specific implementation.

5.8.2.2 SDF data manager

The SDF data manager provides the functionality needed for storing, managing and accessing information in the SDF. If, for example, the data are physically structured as a database, the SDF data manager may also handle database accessing language such as an SQL.

5.8.2.3 Functional entity access manager

The functional entity access manager provides the functionality needed by the SDF data manager to exchange information with other functional entities, i.e., SCF, SDF and SMF, via messages. This message handling functionality should:

- provide reliable message transfer;
- ensure sequential message delivery;
- allow message request/response pairs to be correlated;
- allow multiple messages to be associated with each other;
- comply with OSI structures and principles.

5.8.2.4 Security manager

The Security manager provides secure access to the different types of data held in the SDF, for example, denied access to the data for unauthenticated users. This functionality should:

- check the access rights of the SCF;
- authenticate users with provided information;
- count the failed authentication attempts for a given user;
- block data access;
- assign user's access rights;
- memorize user's access rights during his request;
- control user's right to access specific data.

5.8.3 Data types handled by SDF

The data that are handled by the SDF can be classified into the following types:

- Authentication data - These data are used to authenticate a user that accesses the database through an SCF, e.g., a PIN code, the value of a counter for failed authentication. The set of authentication data used is associated to a level of access rights.
- Operational data - These data are not needed by the SLPIs, but are used by the SDF itself for operational and administrative purposes, e.g., references to an object class, access control data.
- Service data - These data are used for the provision of a service, e.g., a subscriber profile, service provider agreements. These data can be used by several services if necessary.

6 Physical architecture

This clause describes the physical plane of the IN architecture for T1 IN.

The physical plane of the IN conceptual model identifies the different physical entities and the interfaces between these entities.

The physical plane architecture is consistent with the IN conceptual model. The IN conceptual model is a tool that can be used to design the IN architecture to meet the following main objectives:

- service implementation independence;
- network implementation independence;
- vendor and technology independence.

6.1 Requirements and assumptions

6.1.1 Requirements

The key requirements of the physical plane architecture are:

- the functional entities in the T1 IN distributed functional plane can be mapped onto the T1 IN physical entities;
- one or more functional entities may be mapped onto the same physical entity;
- one functional entity cannot be split between two physical entities (i.e. the functional entity is mapped entirely within a single physical entity);
- duplicate instances of a functional entity can be mapped to different physical entities, though not to the same physical entity;
- physical entities can be grouped to form a physical architecture;
- the physical entities may offer standard interfaces;
- vendors must be able to develop physical entities based on the mapping of functional entities and the standard interfaces;
- vendors must be able to support mature technologies and new technologies as they become available.

6.1.2 Assumptions

The following assumptions are made for the development of the physical plane architecture:

- the IN conceptual model is used as a tool to develop the IN physical architecture;
- existing and new technologies can be used to develop the physical entities;
- the specification of functional entities in the distributed functional plane and standard interfaces in the physical plane will make the network vendor independent and service independent;
- for T1 IN, a sufficient number of interfaces will be identified for support of services. Service creation and OAM functions will not be addressed.

6.2 Physical entities (PEs)

This clause describes a selection of PEs to support T1 IN. That selection is not intended to preclude or disallow the application of any other IN PE to support T1 IN.

A. Service switching point (SSP)

In addition to providing users with access to the network and performing any necessary switching functionality, the SSP allows access to the set of IN capabilities. The SSP contains detection capability to detect requests for IN-based services. It also contains capabilities to communicate with other PE(s) containing an SCF, such as a service control point (SCP), and to respond to instructions from the other PE. Functionally, an SSP contains a CCF, an SSF, and, if the SSP is a local exchange, a CCAF. It also may optionally contain an SCF, and/or an SRF, and/or an SDF. The SSP may provide IN-based services to users connected to subtending Network Access Points.

B. Network access point (NAP)

A NAP is a PE that includes only the CCAF and CCF functional entities. It may also be present in the network. The NAP supports early and ubiquitous deployment of IN-based services. This NAP cannot communicate with an SCF, but it has the ability to determine when IN processing is required. It must send calls requiring IN processing to an SSP.

C. Service control point (SCP)

The SCP contains the service logic programs (SLPs) and data that are used to provide IN-based services. The SCP is connected to SSPs by a signaling network. Multiple SCPs may contain the same SLPs and data to improve service reliability and to facilitate load sharing between SCPs. Functionally, an SCP contains an SCF and an SDF. The SCP can access data in an SDP either directly or through a signaling network. The SDP may be in the same network as the SCP, or in another network. The SCP can be connected to SSPs, and optionally to IPs, through the signaling network. The SCP can also be connected to an IP via an SSP relay function.

D. Adjunct (AD)

The Adjunct PE is functionally equivalent to an SCP (i.e., it contains the same functional entities) but it is directly connected to an SSP. Communication between an Adjunct and an SSP is supported by a high-speed interface. This arrangement may result in differing performance characteristics for an Adjunct and an SCP. The application layer messages are identical in content to those carried by the signaling network to an SCP.

An Adjunct may be connected to more than one SSP and an SSP may be connected to more than one Adjunct.

E. Intelligent peripheral (IP)

The IP provides resources such as customized and concatenated voice announcements, voice recognition, and dual tone multi-frequency (DTMF) digit collection, and contains switching matrix to connect users to these resources. The IP supports flexible information interactions between a user and the network. Functionally, the IP contains the SRF. The IP may directly connect to one or more SSPs, and/or may connect to the signaling network.

An SCP or Adjunct can request an SSP to connect a user to a resource located in an IP that is connected to the SSP from which the service request is detected. An SCP or Adjunct can also request the SSP to connect a user to a resource located in an IP that is connected to another SSP.

F. Service node (SN)

The SN can control IN-based services and engage in flexible information interactions with users. The SN communicates directly with one or more SSPs, each with a point-to-point signaling and transport connection. Functionally, the SN contains an SCF, SDF, SRF, and an SSF/CCF. This SSF/CCF is closely coupled to the SCF within the SN, and is not accessible by external SCFs.

In a manner similar to an Adjunct, the SCF in an SN receives messages from the SSP, executes SLPs, and sends messages to the SSP. SLPs in an SN may be developed by the same service creation environment used to develop SLPs for SCPs and Adjuncts. The SRF in an SN enables the SN to interact with users in a manner similar to an IP. An SCF can request the SSF to connect a user to a resource located in an SN that is connected to the SSP from which the service request is detected. An SCF can also request the SSP to connect a user to a resource located in an SN that is connected to another SSP.

G. Service switching and control point (SSCP)

The SSCP is a combined SCP and SSP in a single node. Functionally, it contains an SCF, SDF, CCAF, CCF, and SSF. The connection between the SCF/SDF functions and the CCAF/CCF/SSF functions is proprietary and closely coupled, but it provides the same service capability as an SSP and SCP separately.

The interfaces between the SSCP and other PEs are the same as the interfaces between the SSP and other PEs, and therefore will not be explicitly stated.

H. Service data point (SDP)

The SDP contains the customer and network data that are accessed during the execution of a service. Functionally, the SDP contains an SDF.

6.3 Mapping the distributed functional plane to the physical plane

6.3.1 Mapping of functional entities to physical entities

This clause gives a mapping of functional entities into physical entities for T1 IN, and describes the reference points between the PEs. In so doing, an appropriate distribution of functionality for T1 IN is identified, and functional interfaces suitable for standardization are highlighted. The PEs described in this clause are for illustrative purposes only, and do not imply the only possible mapping of functionality for T1 IN.

This clause describes a flexible physical architecture made up of several PEs. Each PE contains one or more functional entities, which define its IN functionality. PEs included in the physical architecture shown in Figure 19 are SSP, NAP, SCP, IP, AD, SSCP, SDP, and SN.

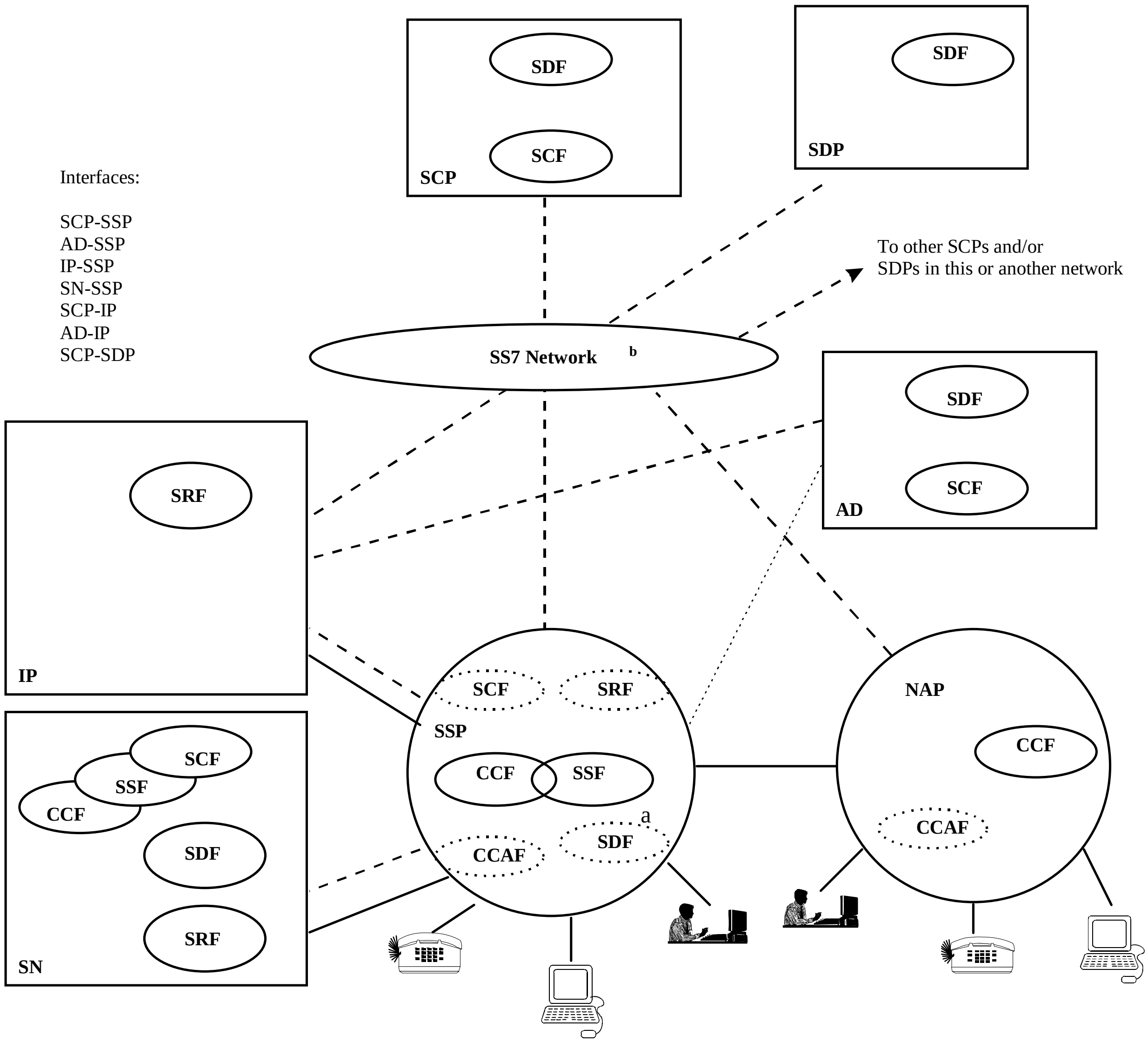
Typical scenarios of functional entity mapping to physical entity are shown in Table 8.

Table 8 - Typical scenarios of FE to PE mapping

PEs	**FEs**			
	SCF	SSF/CCF	SDF	SRF
SSP	O	C	O	O
SCP	C	–	C	–
SDP	–	–	C	–
IP	–	–	–	C
AD	C	–	C	–
SN	C	C	C	C
SSCP	C	C	C	O
NAP	–	C (CCF only)	–	–
C Core O Optional – Not allowed				

There is no intention that the table should disallow any other combination of functional entities that would result in a PE not shown in the table.

The above mappings are shown in Figure 19. Each PE has certain functional entities mapped into it. The solid lines on the figure show transport paths that may exist between the PEs, and the dotted lines show signaling paths that can carry application layer messages for IN-based services.



a) An SSCP PE includes the SCF and
SDF FEs as core elements
b) Other network protocols may be used.

Transport

Signaling

Optional FE

Physical entities (PEs)		Functional entities (FEs)	
AD	Adjunct	CCF	Call control function
IP	Intelligent peripheral	CCAF	Call control agent function
SSP	Service switching point	SCF	Service control function
SCP	Service control point	SDF	Service data function
SN	Service node	SRF	Specialized resource function
NAP	Network access point	SSF	Service switching function
SDP	Service Data point		
SSCP	Service switching and control point		

Figure 19 - Scenarios for physical architecture

6.3.2 Mapping of FE-FE relationships to PE-PE relationships

The FE-FE interfaces that fall within the scope of T1 IN are:

- SSF-SCF;
- SCF-SDF;
- SCF-SRF.

A mapping to the PE-PE interfaces is provided in Table 9.

Table 9 is not meant to be an exhaustive list of all possible PE-PE relationships that may be covered by the T1 IN standard.

Table 9 - FE-FE relationships to PE-PE relationships

FE-FE	PE-PE
SSF-SCF	SSP-SCP SSP-AD SSP-SN
SCF-SDF	SSP-SCP SCP-SDP
SCF-SRF	SCP-IP SCP-SSP-IP AD-IP

6.4 Selection of underlying protocol platforms

This clause describes the candidate interfaces for T1 IN between the elements of the physical architecture. The interfaces are identified below.

- SCP-SSP;
- AD-SSP;
- IP-SSP;
- SN-SSP;
- SCP-IP;
- AD-IP;
- SCP-SDP.

Existing lower-layer protocols are proposed for these candidate interfaces to carry the application layer messages required by IN-based services. As such, the focus of the standardization effort for T1 IN is on the application layer protocols. At the application layer, the message sent that the different interfaces carry should reflect the same semantic content, even though the application layer messages may be encoded or formatted differently. For example, the messages between the SSF in an SSP and the SCF in an SCP, Adjunct or SN should contain the same information. The following clauses give some proposed protocols for use on these interfaces.

6.4.1 SCP-SSP interface

The proposed underlying protocol platform for the interface between an SCP and an SSP is transaction capabilities application part (TCAP) on signaling connection control part (SCCP)/message transfer part (MTP) of SS7.

6.4.2 AD-SSP interface

The proposed underlying protocol platform for the AD-SSP interface is TCAP. The physical interface has not been specified, but a number of alternative standard protocols could be used.

6.4.3 IP-SSP interface

This interface is used for communications between an IP and an SSP as well as for communication between an IP and an SCP that is being relayed through an SSP.

The proposed underlying protocol platform for the interface between an IP and an SSP is ISDN basic rate interface (BRI), primary rate interface (PRI) (or both), or SS7.

If a BRI or PRI is used, the ISDN D-channel connecting an IP to an SSP carries application layer information between an SCF and an SRF, and supports the setup of B channel connections to the IP. Information passed from an SCF to an SRF (e.g., announcement number and number of digits to collect) and vice versa (e.g., collected information and billing measurements) is embedded in the facility information element. The facility information element can be carried by some T1.607 messages, such as SETUP and DISCONNECT. The facility information element can also be carried by the FACILITY message of T1.607. This possibility provides for the flexibility to convey application layer information without affecting the call connection establishment.

6.4.4 SN-SSP interface

The proposed underlying protocol platform for the interface between an SN and an SSP is ISDN basic rate interface (BRI), primary rate interface (PRI) (or both). An SN and an SSP exchange application layer messages over an ISDN D-channel using the common element procedures of T1.610. This communication may occur on a separate D-channel from the channel that carries the common element procedure messages. Figure 19 shows the case where these channels are separate.

6.4.5 SCP-IP interface

The proposed underlying protocol platform for an interface between an SCP and an IP is transaction capabilities application part (TCAP) on signaling connection control part (SCCP)/MTP of SS7 or TCAP on Broadband Signaling Transport.

6.4.6 AD-IP interface

The proposed underlying protocol platform between an AD and an IP is TCAP. The physical interface has not been specified, but a number of alternative standard protocols could be used.

6.4.7 SCP-SDP interface

The proposed underlying protocol platform for the interface between an SCP and an SDP is transaction capabilities application part (TCAP) in signaling connection control part (SCCP)/MTP of SS7.

6.4.8 User interfaces

A user is an entity external to the IN that uses IN capabilities. IN users may employ the access interfaces described below to invoke various IN service capabilities. For example, users can affect the routing of a call, send and receive information from the network, screen calls, and update service parameters. Users are served by existing network interfaces.

It is important to ensure that IN should continue to support existing services and capabilities. In addition, the current restrictions imposed by each of the interface technologies described below must be considered when deploying IN-based services. For example, calling party information may or may not be available at a given interface and, therefore, may or may not be provided to the SCF.

End users use analog interface signaling, or ISDN access signaling arrangements. IN user-network interactions include providing stimuli, such as off-hook or DTMF digit signaling, which determine further IN action.

Out-of-band (i.e., D-channel) signaling provides ISDN users with additional capabilities for accessing potential IN-based services. When originating a call, an ISDN user identifies the bearer capability to be associated with the call. IN service logic can use this information to determine how the call should be handled (e.g., how to route the call).

7 IN Application Protocol (INAP)

7.1 Introduction

This clause defines the INAP (intelligent network application protocol) required for support of the T1 IN Standard. It supports interactions between the following four functional entities (FE’s) as defined in the IN functional model:

- Service switching function (SSF)
- Service control function (SCF)
- Specialized resource function (SRF)
- Service data function (SDF)

The scope of this Standard is the further development of the INAP for both the Integrated Services Digital Network (ISDN) and Public Switched Telecommunications Network (PSTN).

It is intended as a guide to implementers and network operators to ensure interworking between different manufacturers equipment for all the T1 IN defined interfaces (SCF-SSF, SCF-SRF and SCF-SDF) and between network operators for the internetwork interface (SCF-SDF). See Annex C for the description of the SCF-SDF interface.

The protocol in the T1 IN standard is based upon the ITU-T IN CS-1 Recommendations (1995) with adaptations to support North American requirements and implementations. The adaptations may range from simple name changes to more extensive redefinitions (e.g., different parameters, somewhat different functionality). In some cases, an operation may be used in T1 IN exactly as specified by the ITU-T with the adaptation confined to what is required to ensure support within North American protocols (e.g., private TCAP). In other cases, an operation may be re-specified for T1 IN. An adaptation of this type is necessary in some cases to support American implementations that pre-date the ITU-T Recommendations.

7.1.1 Definition methodology

The definition of the protocol is split into three clauses:

- the definition of the SACF/MACF rules for the protocol (see 7.2);
- the definition of the operations transferred between entities (see 7.3);
- the definition of the actions taken at each entity (see 7.4).

The SACF/MACF rules are defined in prose. The operation definitions are in Abstract Syntax Notation 1 (ASN.1, see ITU-T Recommendation X.680), and the actions are defined in terms of state transition diagrams. Further guidance on the actions to be performed on receipt of an operation can be gained from 7.4.3.

The INAP uses the concepts of the Remote Operation Service (ROS) (see ITU-T Recommendation X.880). The ROSE protocol is contained within TCAP (see T1.114) and DSS1 (T1.607). Annex A defines the mapping from INAP operations to the component portion of TCAP messages in SS7 and Facility Information Elements in the T1.607 REGISTER, FACILITY and call control messages in DSS1. Other supporting protocols may be added at a later date.

The INAP (as a ROSE user) and the ROSE protocols have been specified using ASN.1. At present, the only standardized way to encode the resulting PDUs is the Basic Encoding Rules (see ITU-T Recommendation X.690). Also, note that it is necessary to consult T1.114.3 for detailed TCAP data element encoding specifications as application of basic encoding rules to an ASN.1 specification does not provide a complete message format specification.

7.1.2 Example physical scenarios

The protocol will support any mapping of functional to physical entities (PE’s). It is the responsibility of network operators and equipment manufacturers to decide how to co-locate FE’s to the best possible advantage as this may vary between manufacturers and between network operators. Therefore the protocol is defined assuming maximum distribution (i.e. one PE per FE).

The figures depicted in this clause show how INAP would be supported in an SS7 network environment. This does not imply that only SS7 may be used as the network protocol to support INAP.

The interface, as depicted in Figure 20, between remotely located SCF and SDF will be INAP using TCAP which in turn, uses the services of the connectionless SCCP and MTP. The SDF is responsible for any interworking to other protocols to access other types of networks.

When TCAP appears in one of the following figures, it shall be understood as representing the TCAP functionalities associated with a single TCAP transaction (as opposed to a TCAP entity).

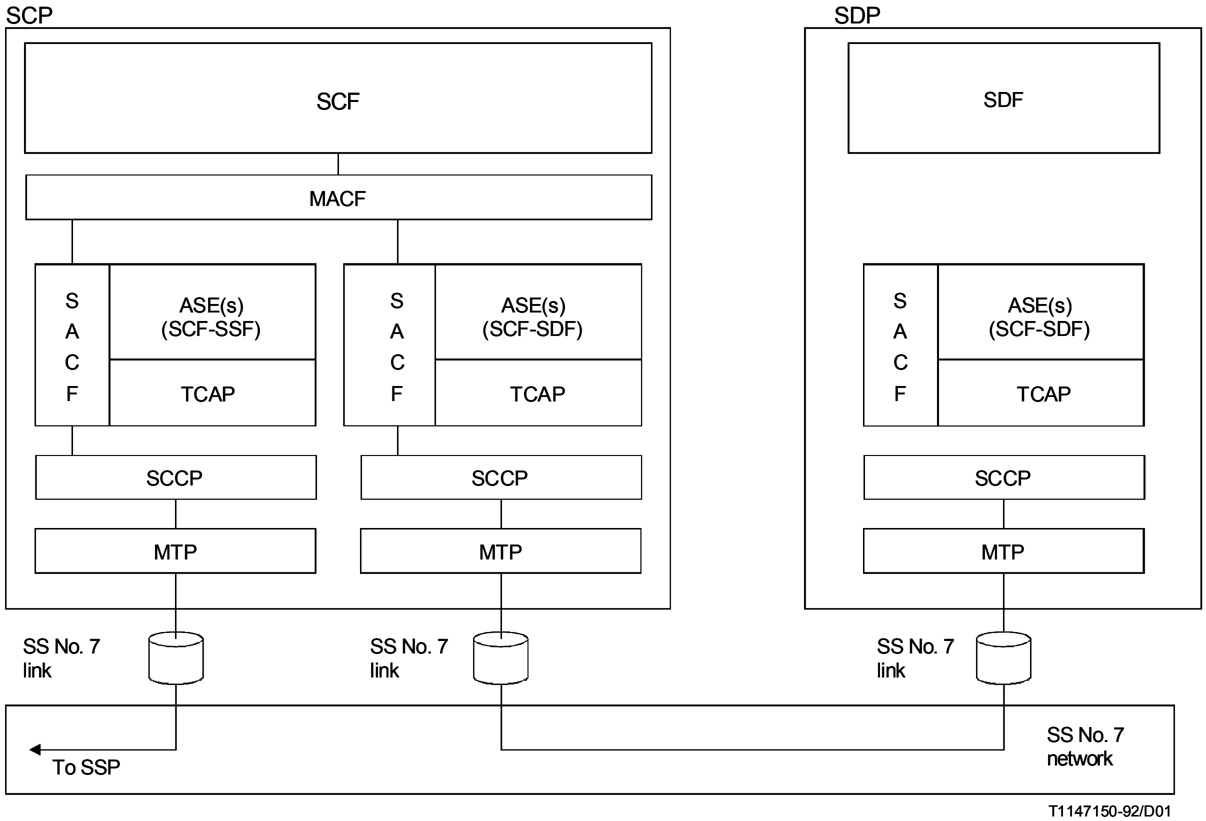
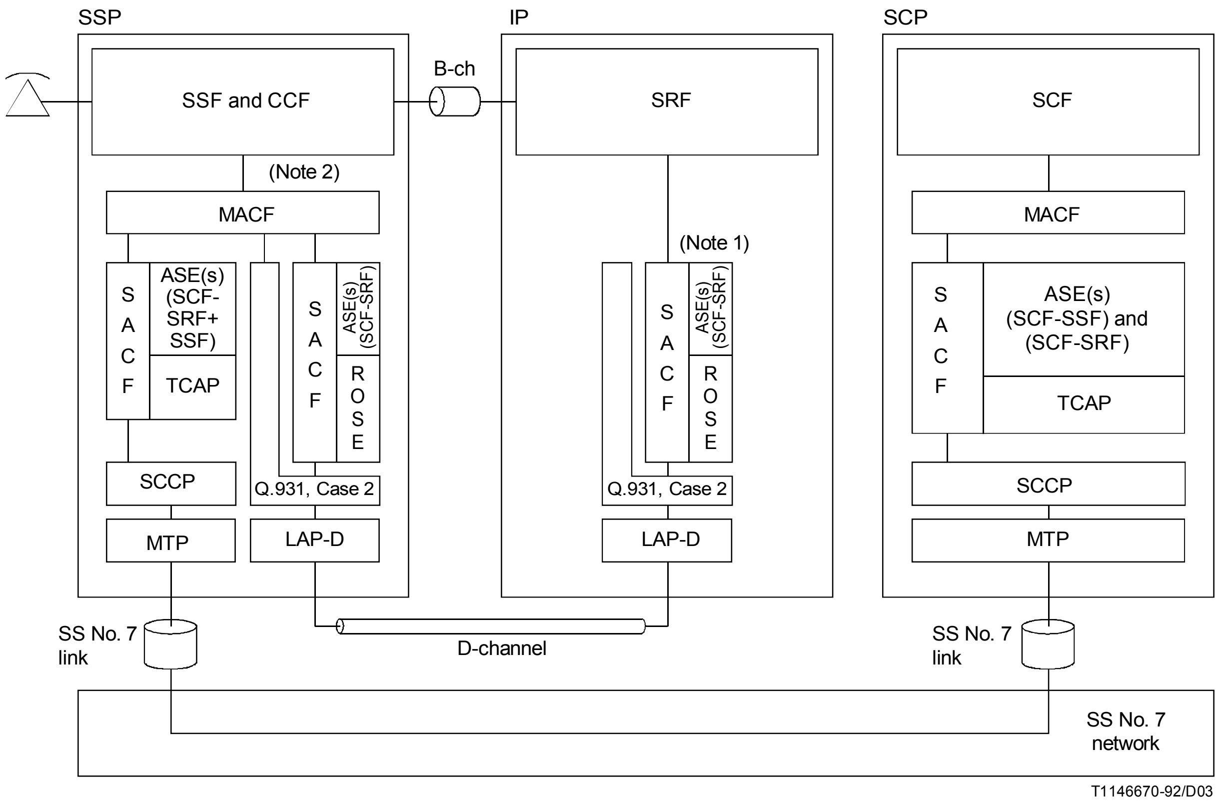


Figure 20 - Physical interface between SCP and SDP

If segmentation and re-assembly of INAP messages are required on the SCF-to-SDF interface (and on other interfaces, if needed) due to the length of messages, the segmentation and re-assembly procedure for SCCP connectionless messages, as specified in T1.112.4, should be used.

A number of example scenarios have been identified for support of the SCF, SSF and SRF functional entities as physical entities. These are illustrated in Figure 21, Figure 22, and Figure 23. Each example is characterized by:

- the method to support the SCF-SRF relationship; and
- the type of signaling system between the SSF and SRF.



NOTES

1 Info flows between SCF and SRF are supported by this (ROSE) entity.

2 Relay function is provided either by MACF or by application process at SSP.

Figure 21 - Example architecture for supporting SRF (SRF in IP connected to SSP and accessed by SCP through D-channel via SSP)

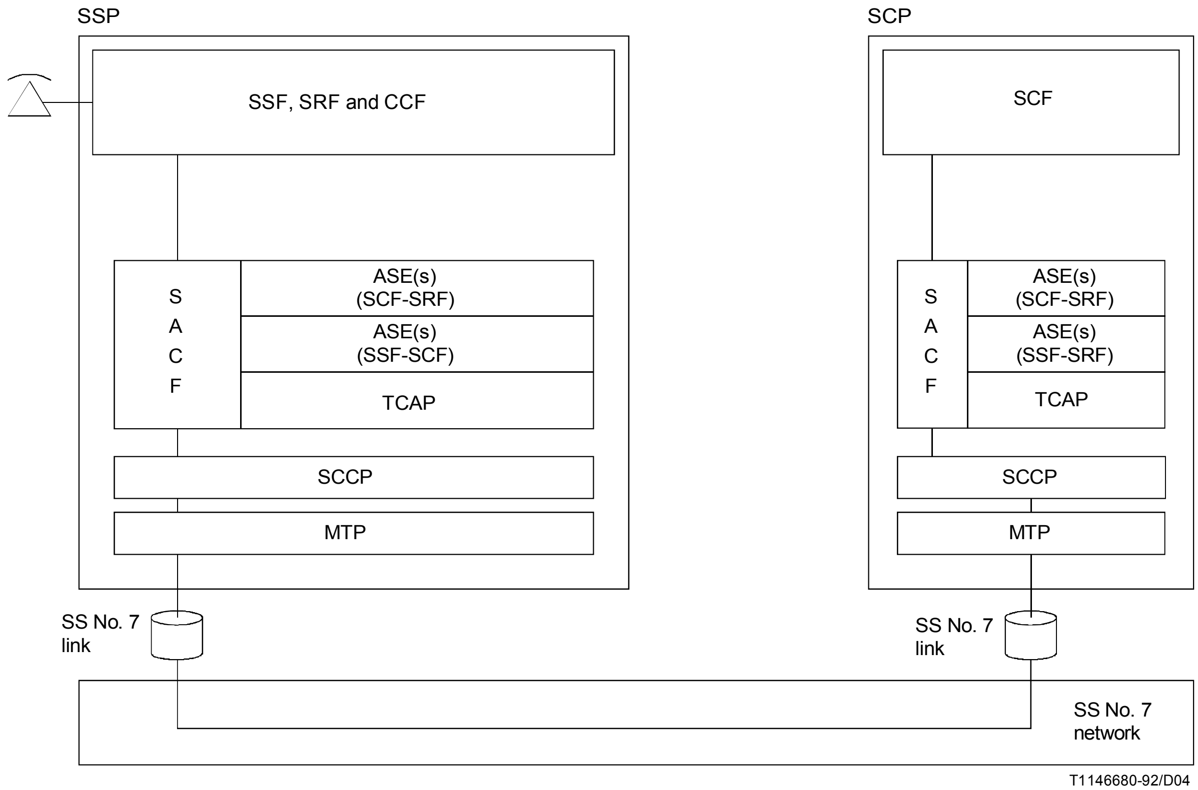
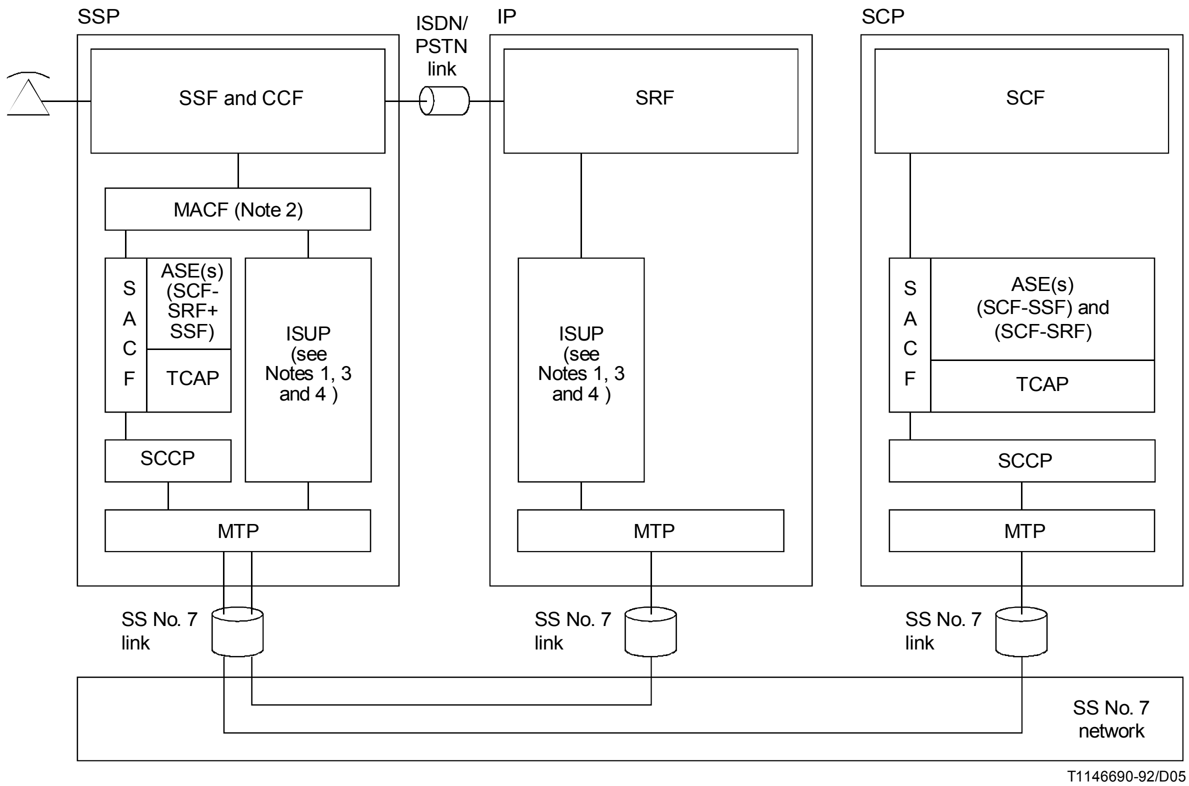


Figure 22 - Example architecture for supporting SRF (SRF in SSP and accessed via AP of SSP)



NOTES

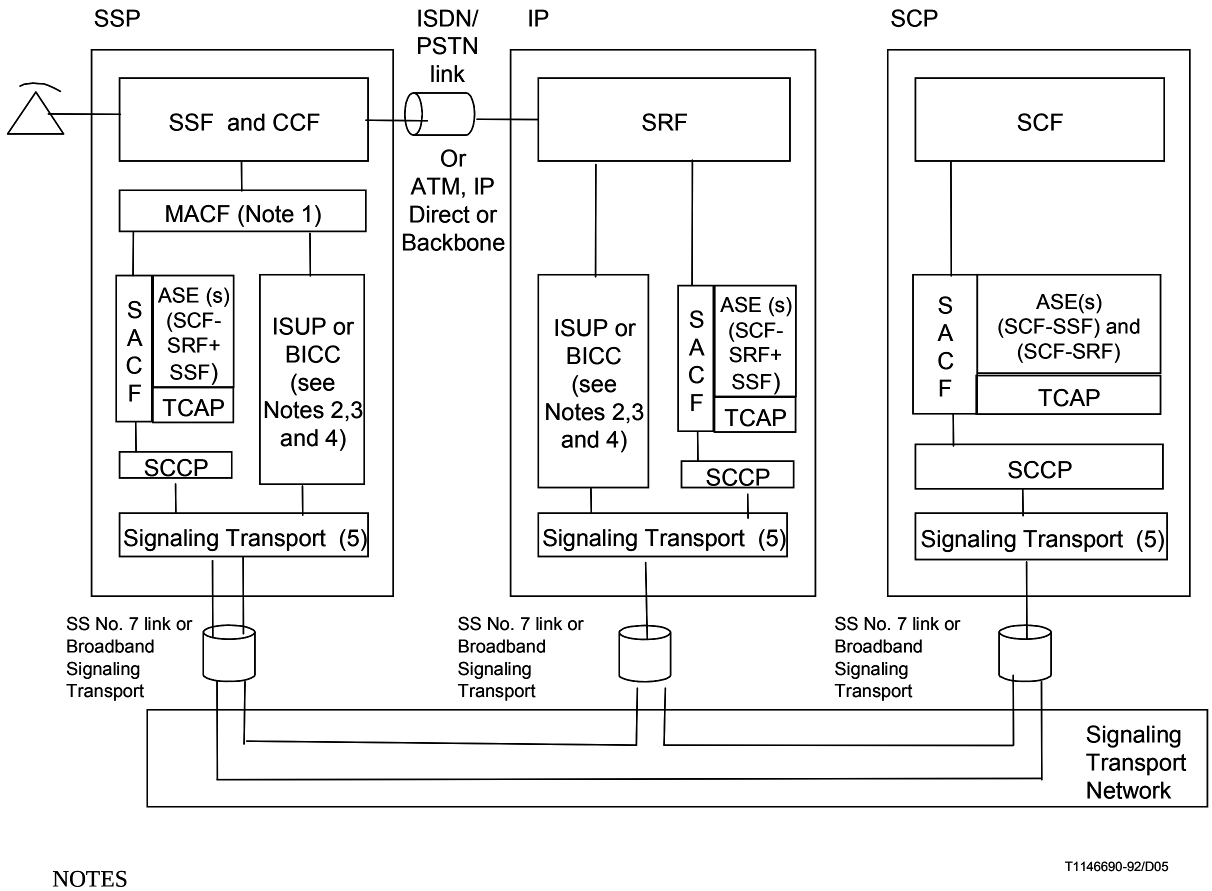
1 Info flows between SCF and SRF as well as connection control are directly supported by ISUP.

2 Relay function is provided either by MACF or by application process at SSP.

3 Assumes that ISUP provides a means to transport ROSE information.

4 Other signalling systems may be used.

Figure 23 - Example architecture for supporting SRF
(SRF in IP connected to SSP and accessed by SCP through ISUP via SSP)



NOTES

1 Relay function is provided either by MACF or by application process at SSP.
2 Assumes that ISUP/BICC provides a means to transport ROSE information.
3 Other signalling systems may be used.
4 There is an Signaling Transport Converter between BICC and the signaling transport being used.
5 The initial connection is set through the SSF and the subsequent communication is directly between the SCF and the SRF.

Figure 23b - Example architecture for supporting SRF
(SRF in IP connected to SSP and accessed directly by SCP)

Table 10 summarizes the selection of features for Figure 21, Figure 22, Figure 23, and Figure 23b.

Table 10 - SCF-SRF Relationship

Type of signaling system between SSF and SRF	Method to support SCF-SRF relationship	
	Direct TCAP link	**Relay via SSP**
ISUP	Figure 23b [d)]	Figure 23[c)]
DSS1	Figure 23b [d)]	Figure 21[a)]
Implementation dependent	Not Selected	Figure 22[b)]

Additional information related to each figure:

a) Figure 21: IP can be accessed by DSS1 only. The IP can be a physical entity residing outside the network.

b) Figure 22: SSP supports both CCF/SSF and SRF. The handling of SRF by SCF could be the same as the of Figure 21.

c) Figure 23: IP can be accessed by ISUP only. The handling of SRF by SCF could be the same as that of Figure 21.

d) Figure 23b: IP can be accessed by DSS1 or ISUP. The Direct TCAP link can be supported by SS No. 7 link or Broadband Signaling Transport.

7.1.3 INAP protocol architecture

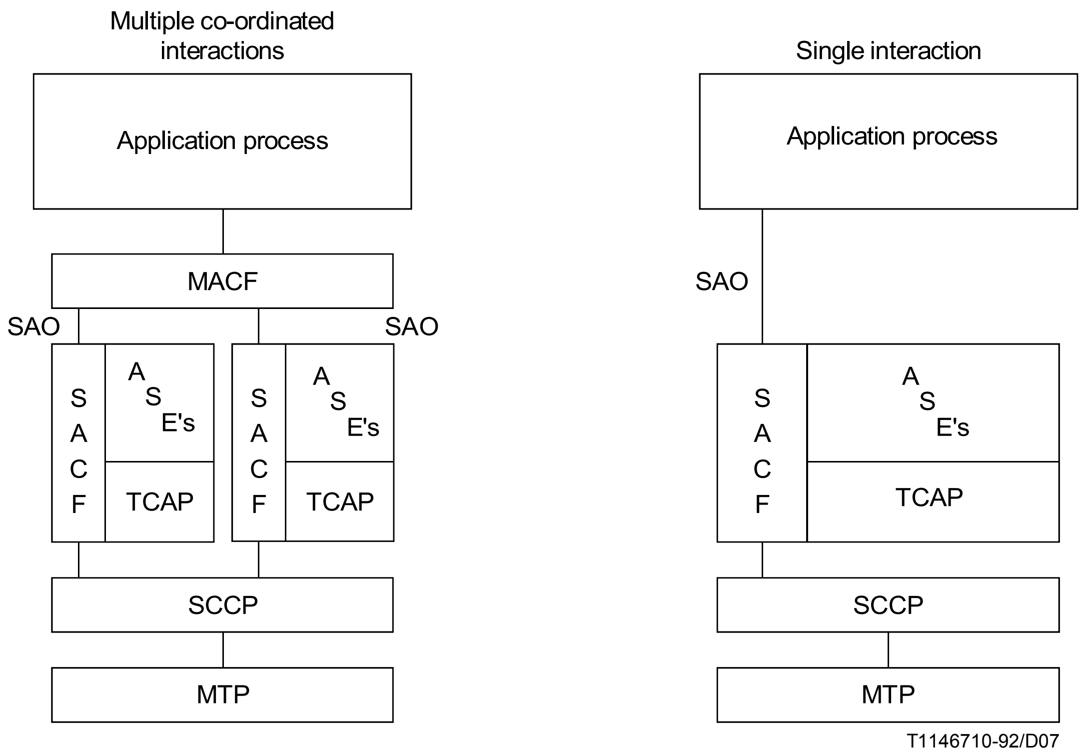
The INAP protocol architecture can be illustrated as shown in Figure 24.

A physical entity has either single interactions or multiple coordinated interactions with other physical entities.

For single interactions, the SACF provides a coordination function in using ASEs, which includes the ordering of operations supported by ASE(s), (based on the order of received primitives). The SAO represents the SACF plus a set of ASEs to be used over a single interaction between a pair of PE's.

For multiple coordinated interactions, the MACF provides a coordinating function among several SAO's, each of which interacts with an SAO in a remote PE.

Each ASE supports one or more operations. Description of each operation is tied with the action of corresponding FE modeling (see clause 5 of this Standard). Each operation is specified as an instance of the OPERATION information object class described in Figure 25.



SACF	Single association control function
MACF	Multiple association control function
SAO	Single association object
ASE	Application service element
INAP	Intelligent network application protocol

Figure 24 - INAP protocol architecture

The use of the application context negotiation mechanism [as defined in T1.114.4 (*Transaction Capabilities Procedures*)] allows the two communicating entities to identify exactly what their capabilities are and also what the capabilities required on the interface should be. This should be used to allow evolution through Intelligent Network capability sets.

If the indication of a specific application context is not supported by a pair of communicating FE's, some mechanism to pre-arrange the context must be supported.

7.1.4 INAP addressing

SCCP global title and MTP point code addressing [see T1.112 (*Signaling connection control part*) and T1.111 (*Message transfer part*)] ensure that PDUs reach their physical destination (i.e., the correct point code) regardless of in which network that destination is situated.

Within a node, it is the choice of the network operator/implementer as to which SSN or SSNs are assigned to INAP.

Regardless of the above, any addressing scheme supported by the SCCP may be used.

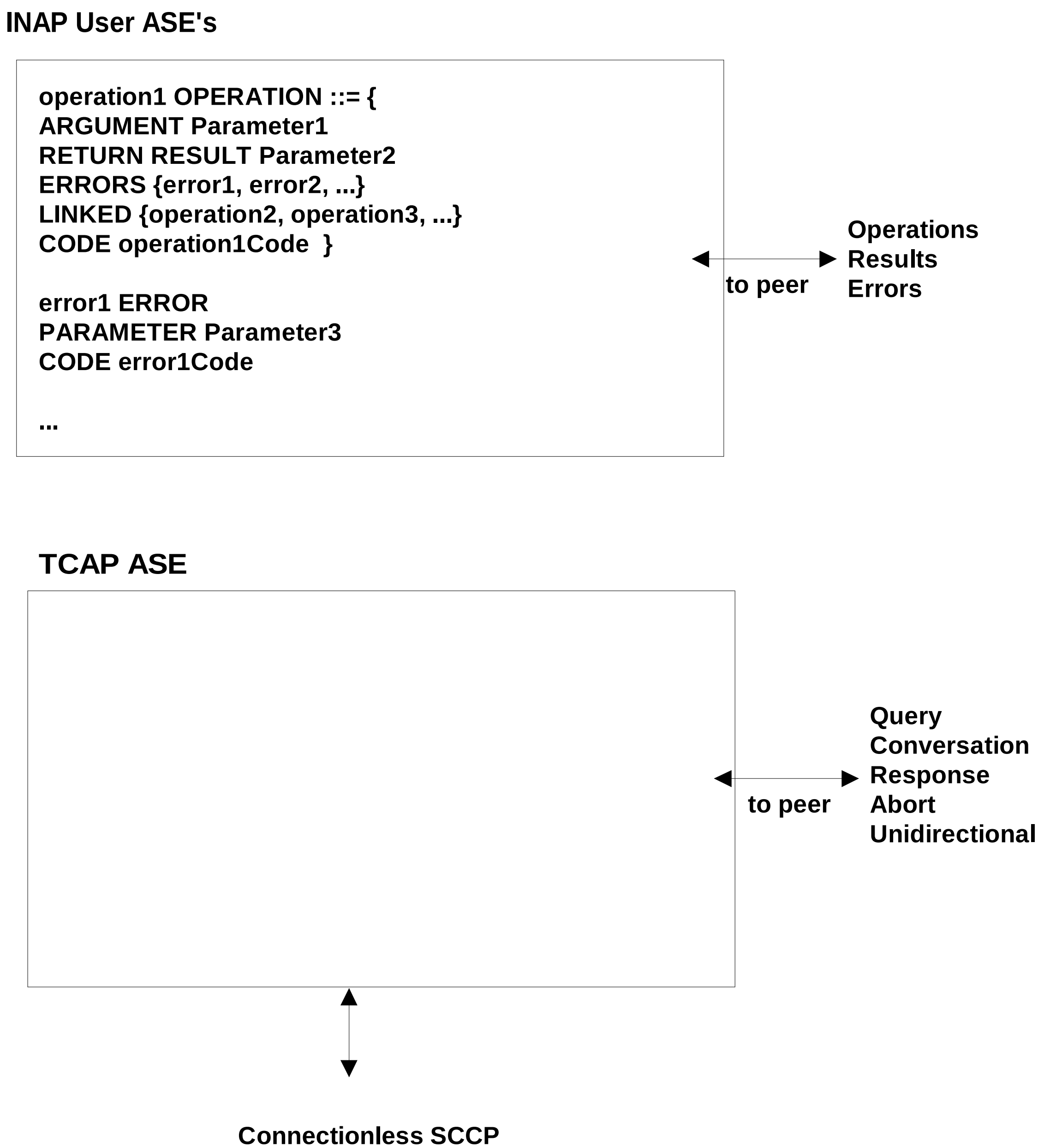


Figure 25 - Operation description

7.1.5 Compatibility mechanisms used for INAP

7.1.5.1 Introduction

This clause specifies the compatibility mechanisms that shall be used to ensure consistent future versions of INAP.

There are three categories of compatibility:

- Minor changes to INAP in future standardized versions:
  A minor change can be defined as a change of a functionality that is not essential for the requested IN service. In case it is a modification of an existing function, it is acceptable that the addressed function is executed in either the older or the modified variant. If the change is purely additional, it is acceptable that it is not executed at all and that the peer Application Entity (AE) need not know about the effects of the change. For minor changes, a new AC is not required.
- Major changes to INAP in future standardized versions:
  A major change can be defined as a change of a functionality that is essential for the requested IN service. In case it is a modification of an existing function, both application entities shall have a shared knowledge about the addressed functional variant. If the change is purely additional, the requested IN service will not be provided if one of the application entities does not support the additional functionality. For major changes, a new AC is required.
- Network-specific changes to INAP:
  These additions may be of either the major or minor type for a service. No new AC is expected to be defined for this type of change.

7.1.5.2 Definition of INAP compatibility mechanisms

7.1.5.2.1 Procedures for major additions to INAP

In order to support the introduction of major functional changes, the protocol allows a synchronization between the two applications with regard to which functionality is to be performed. This synchronization takes place before the new function is invoked in either application entity, in order to avoid complicated fall-back procedures. The solution chosen to achieve such a synchronization is to use the AC negotiation procedures provided in T1.114.4.

7.1.5.2.2 Procedures for minor additions to INAP

The extension mechanism marker shall be used for future standardized minor additions to INAP. This mechanism implements extensions differently by including an “extensions marker” in the type definition. The extensions are expressed by optional fields that are placed after the marker. When an entity receives unrecognized parameters that occur after the marker, they are ignored.

7.2 SACF/MACF rules

7.2.1 Reflection of TCAP AC

TCAP Application Context negotiation rules require that the proposed AC, if acceptable, is reflected in the first backwards message.

If the AC is not acceptable, and the TC-User does not wish to continue the dialogue, it may provide an alternate AC to the initiator that can be used to start a new dialogue.

TCAP AC negotiation applies only to the SCF interfaces.

Refer to the T1.114.4 (*Transaction capabilities application part*) for a more detailed description of the TCAP AC negotiation mechanism.

7.2.2 Sequential/Parallel execution of operations

In some cases, it may be necessary to distinguish whether operations should be performed sequentially or in parallel (synchronized). Operations that may be synchronized are:

- charging operations may be synchronized with any other operation.

The method of indicating that operations are to be synchronized is to include them in the same message. Where one of the operations identified above must not be executed until some other operation has progressed to some extent or finished, the sending PE (usually SCP) can control this by sending the operations in two separate messages.

This method does not imply that all operations sent in the same message should be executed simultaneously, but simply that where it could make sense to do so (in the situations identified above) the operations should be synchronized.

In case of inconsistency between the above-mentioned generic rules and the FE-specific rules, as specified in 7.4, the FE-specific rules take precedence over the generic rules.

7.3 Abstract syntax of the T1 IN application protocol

This clause specifies the abstract syntax for the T1 IN Application Protocol using Abstract Syntax Notation One (ASN.1), defined in ITU-T Recommendation X.680.

The encoding rules that are applicable to the defined abstract syntax are the basic encoding rules for ASN.1, defined in ITU-T Recommendation X.690 with the restrictions as described in section 4.1.1 of ITU-T Q.773. Additional encodings are cited for parameters used in existing ISUP (T1.113) and DSS1 (T1.607/T1.610) ANSI Standards.

For ISUP parameters and DSS1 information elements used in INAP, the INAP parameter type is OCTET STRING. The format in T1.113 (ISUP parameters) or T1.607/T1.610 (DSS1 information elements) determines the internal structure of this octet string. Application of BER to the INAP ASN.1 definition determines the parameter identifiers; they are not ISUP parameter name codes or DSS1 information element identifiers.

The mapping of OPERATION and ERROR to TCAP components is defined in Annex A. The class of an operation is not stated explicitly but is specified in the ASN.1 OPERATION MACRO, as follows:

- Class 1 Both RESULT and ERRORS appear in the ASN.1 OPERATION definition.
- Class 2 Only ERRORS appears in the ASN.1 OPERATION definition.
- Class 3 Only RESULT appears in the ASN.1 OPERATION definition.
- Class 4 Neither RESULT nor ERRORS appears in the ASN.1 OPERATION definition.

The abstract syntax for INAP is composed of several ASN.1 modules describing operations, errors, and associated data types. The values (operation codes and error codes) are defined in a separate module.

The module containing all the type definitions for INAP operations is T1 IN Operations (See 7.3.1.1 and 7.3.2.2.5.2).

The module containing all the type definitions for INAP errors is T1 IN Errors (See 7.3.1.2 and 7.3.2.2.5.2).

The module containing all the type definitions for INAP data types is T1 IN Data Types (See 7.3.1.3 and 7.3.2.2.5.2).

The modules containing the SSF/SRF-SCF Packages, Contracts, and Application Contexts are ANSI-IN-SSF-SCF-pkgs-contracts-acs, ANSI-IN-ITU-SSF-SCF-pkgs-contracts-acs, and ANSI-IN-ITU-SCF-SRF-pkgs-contracts-acs (See 7.3.1.4, 7.3.1.5, 7.3.1.6, and 7.3.2.2.5.2).

7.3.1 SSF-SCF, SCF-SRF interface

7.3.1.1 T1 IN operation types

T1 IN Operations {iso(1) memberbody(2) us(840) t1 667 2002(10069) modules(0) inap(0) version1(0)}

-- This module contains the type definitions for the T1 IN operations.
-- There may be functional redundancies in the operation set related to call processing.
-- This may make product interworking more difficult. Administrations wishing to deploy
-- IN and equipment manufacturers implementing IN should take this into account.

DEFINITIONS ::=

BEGIN

IMPORTS

OPERATION

FROM IN-Remote-Operations-Information-Objects {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) informationObjects(5) version1(0)}

-- error types
applicationError,
Canceled,
CancelFailed,
failureReport,
ImproperCallerResponse,
MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
RequestedInfoError,
SystemFailure,
TaskRefused,
UnavailableResource,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter,
UnknownLegID,
UnknownResource

FROM ANSI-IN-Errors {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inaperrors(1) version1(0)}

-- argument types
AcgArg,
ActivateServiceFilteringArg,
AnalyzedInformationArg,
AnalyzeRouteArg,
ApplyChargingArg,
ApplyChargingReportArg,
AssistRequestInstructionsARG,
AuthorizeTerminationArg,
CallInfoFromResourceArg,
CallInfoFromResourceRR,
CallInformationReportArg,

```
        CallInformationRequestArg,
        CancelResourceEventArg,
        CloseArg,
        CollectedInformationArg,
        CollectInformationArg,
        ContinueArg,
        EventNotificationChargingArg,
        FacilitySelectedAndAvailableArg,
        ForwardCallArg,
        FurnishChargingInformationArg,
        InstructionsToSRFArg,
        MidCallArg,
        MonitorForChangeArg,
        MonitorSuccessArg,
        NCADAtaArg,
        NCARequestArg,
        OAnswerArg,
        OCalledPartyBusyArg,
        ODisconnectArg,
        ODTMFEnteredArg,
        OfferCallArg,
        ONoAnswerArg,
        OriginationAttemptArg,
        OSuspendedArg,
        OTermSeizedArg,
        ReleaseCallArg,
        ReportErrorArg,
        RequestNotificationChargingEventArg,
        RequestReportBCMEventArg,
        ResetTimerArg,
        ResourceClearArg,
        RouteSelectFailureArg,
        SendChargingInformationArg,
        SendNotificationArg,
        SendToResourceArg,
        ServiceFilteringResponseArg,
        SpecializedResourceReportArg,
        StatusReportedArg,
        TAnswerArg,
        TBusyArg,
        TerminationAttemptArg,
        TerminationNotificationArg,
        TimeoutArg,
        TNoAnswerArg,
        UpdateArg,
        UpdateRequestArg

FROM ANSI-IN-DataTypes {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inapdatatypes(2)
version1(0)}
-- TYPE DEFINITIONS FOR T1 IN OPERATIONS FOLLOWS
-- SCF-SSF operations
acg                 OPERATION ::= {
ARGUMENT            AcgArg
RETURN RESULT       FALSE
ERRORS              {applicationError
                    }
CODE                opcode-acg
}
opcode-acg ::= private: 26881
-- Family Name is “Network Management”
```

```
-- Direction: SCF -> SSF.
-- This operation is used to request the SSF to reduce the rate at which specific service requests are sent to
-- the SCF.

activateServiceFiltering        OPERATION ::= {
ARGUMENT                        ActivateServiceFilteringArg
RESULT
ERRORS                          {MissingParameter|
                                ParameterOutOfRange|
                                SystemFailure|
                                TaskRefused|
                                UnexpectedComponentSequence|
                                UnexpectedParameter
                                }
CODE                            opcode-activateServiceFiltering
}
opcode-activateServiceFiltering ::= private: 32513
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF –> SSF, Timer: Tasf
-- When receiving this operation, the SSF handles calls to destination in a specified manner
-- without sending queries for every detected call. It is used for example for providing
-- televoting or mass calling services. Simple registration functionality (counters) and
-- announcement control may be located at the SSF. The operation initializes the specified
-- counters in the SSF.

activityTest              OPERATION ::= {
RESULT
CODE                      opcode-activityTest
}
opcode-activityTest ::= private: 32514
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF –> SSF, Timer: Tat
-- This operation is used to check for the continued existence of a relationship between the SCF
-- and SSF. If the relationship is still in existence, then the SSF will respond. If no reply is
-- received, then the SCF will assume that the SSF has failed in some way and will take the
-- appropriate action.

analyzedInformation       OPERATION ::= {
ARGUMENT                  AnalyzedInformationArg
RETURN RESULT             FALSE
ERRORS                    {applicationError |
                          FailureReport
                          }
CODE                      opcode-analyzedInformation
}
opcode-analyzedInformation ::= private: 25603
-- Family Name is "Request/Notification"
-- Direction:  SSF –> SCF, Timer: Tadi
-- This operation is used to indicate availability of routing address and call type. (DP Analyzed_Information).

analyzeRoute              OPERATION ::= {
ARGUMENT                  AnalyzeRouteArg
RETURN RESULT             FALSE
ERRORS                    {applicationError
```

```
                        }
CODE                    opcode-analyzeRoute
}
opcode-analyzeRoute ::= private: 25857
-- Family Name is "Connection Control (Call Model)"
-- Direction: SCF -> SSF.
-- This operation is used to request the SSF to resume call origination processing at the
-- Analyze_Information or Select_Route PICs.

applyCharging           OPERATION ::= {
ARGUMENT                ApplyChargingArg
RETURN RESULT           FALSE
ERRORS                  {MissingParameter|
                        UnexpectedComponentSequence|
                        UnexpectedParameter|
                        UnexpectedDataValue|
                        ParameterOutOfRange|
                        SystemFailure|
                        TaskRefused
                        }
CODE                    opcode-applyCharging
}
opcode-applyCharging ::= private: 32515
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF –> SSF, Timer: Tac
-- This operation is used for interacting from the SCF with the SSF charging mechanisms.
-- The ApplyChargingReport operation provides the feedback from the SSF to the SCF.

applyChargingReport     OPERATION ::= {
ARGUMENT                ApplyChargingReportArg
RETURN RESULT           FALSE
ERRORS                  {MissingParameter|
                        UnexpectedComponentSequence|
                        UnexpectedParameter|
                        UnexpectedDataValue|
                        ParameterOutOfRange|
                        SystemFailure|
                        TaskRefused
                        }
CODE                    opcode-applyChargingReport
}
opcode-applyChargingReport ::= private: 32516
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SSF –> SCF, Timer: Tacr
-- This operation is used by the SSF to report to the SCF the occurrence of a specific charging event as
-- requested by the SCF using the ApplyCharging operation.

assistRequestInstructions       OPERATION ::= {
ARGUMENT                        AssistRequestInstructionsArg
RETURN RESULT                   FALSE
ERRORS                          {applicationError |
                                FailureReport
                                }
```

```
CODE                        opcode-assistRequestInstructions
}
opcode-assistRequestInstructions ::= private: 32521
-- Family Name is "T1 IN ITU operations"
-- Direction: SRF –> SCF, Timer: Tari
-- This operation is used by the SRF to obtain instructions directly from the SCF.

authorizeTermination    OPERATION ::= {
ARGUMENT                AuthorizeTerminationArg
RETURN RESULT           FALSE
ERRORS                  {applicationError
                        }
CODE                    opcode-authorizeTermination
}
opcode-authorizeTermination ::= private: 25858
-- Family Name is "Connection Control (Call Model)"
-- Direction: SCF -> SSF.
-- This operation is used to request the SSF to resume call processing at the
-- Authorize_Termination_Attempt PIC.

callInfoFromResource    OPERATION ::= {
ARGUMENT                CallInfoFromResourceArg
RETURN RESULT           CallInfoFromResourceRR
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-callInfoFromResource
}
opcode-callInfoFromResource ::= private:  26116
-- Family Name Is "Caller Interaction"
-- Direction: SSF -> SCF or SRF -> SCF.
-- This operation is used to send information from the SSF/SRF to the SCF.

callInformationReport   OPERATION ::= {
ARGUMENT                CallInformationReportArg
RETURN RESULT           FALSE
CODE                    opcode-callInformationReport
}
opcode-callInformationReport ::= private: 32517
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SSF   SCF, Timer: Tcirp
-- This operation is used to send specific call information for a single call to the SCF as requested by the
-- SCF in a previous CallInformationRequest.

callInformationRequest      OPERATION ::= {
ARGUMENT                    CallInformationRequestArg
RETURN RESULT               FALSE
ERRORS                      {MissingParameter|
                            ParameterOutOfRange|
                            RequestedInfoError|
                            SystemFailure|
                            TaskRefused|
                            UnexpectedComponentSequence|
```

```
                                        UnexpectedDataValue|
                                        UnexpectedParameter
                                        }
CODE                                    opcode-callInformationRequest
}
opcode-callInformationRequest ::= private: 32518
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF  SSF, Timer: Tcirq
-- This operation is used to request the SSF to record specific information about a single call and report it
-- to the SCF (with a CallInformationReport operation).

cancelResourceEvent         OPERATION ::= {
ARGUMENT                    CancelResourceEventArg
RETURN RESULT               FALSE
ERRORS                      {applicationError
                            }
CODE                        opcode-cancelResourceEvent
}
opcode-cancelResourceEvent ::= private: 26115
-- Family Name is "Caller Interaction"
-- Direction: SCF -> SSF.
-- This operation is used to disconnect a call party from a resource (e.g., an announcement).

close                   OPERATION ::= {
ARGUMENT                CloseArg
RETURN RESULT           FALSE
ERRORS                  {applicationError
                        }
CODE                    opcode-close
}
opcode-close ::= private:  28161
-- Family Name is "Transaction Control"
-- Direction: SCF -> SSF.
-- This operation is used to close an existing transaction.

collectedInformation    OPERATION ::= {
ARGUMENT                CollectedInformationArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-collectedInformation
}
opcode-collectedInformation ::= private: 25602
-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer: Tcdi
-- This operation is used to indicate availability of complete initial information package/dialing string from
-- originating party. (This event may have already occurred in the case of en bloc signaling, in which case
-- the waiting duration in this PIC is zero.) (DP Collected_Information).

collectInformation      OPERATION ::= {
ARGUMENT                CollectInformationArg
RETURN RESULT           FALSE
```

```
ERRORS                  {applicationError
                        }
CODE                    opcode-collectInformation
}
opcode-collectInformation ::= private:  25861
-- Family Name is  "Connection  Control  (Call  Model)"
-- Direction: SCF –> SSF, Timer: Tci
-- This operation is used to request the SSF to perform the originating basic call processing actions to
-- prompt a calling party for destination information, then collect destination information according to a
-- specified numbering plan (e.g. for virtual private networks).

continue                OPERATION ::= {
ARGUMENT                ContinueArg
RETURN RESULT           FALSE
ERRORS                  {applicationError
                        }
CODE                    opcode-continue
}
opcode-continue ::= private: 25869
-- Family Name is "Connection Control (Call Model)"
-- Direction: SCF –> SSF, Timer: Tcue
-- This operation is used to request the SSF to proceed with call processing at the DP at which it
-- previously suspended call processing to await SCF instructions (i.e. proceed to the next point
-- in call in the BCSM). The SSF continues call processing without substituting new data from SCF.

eventNotificationCharging       OPERATION ::= {
ARGUMENT                        EventNotificationChargingArg
RETURN RESULT                   FALSE
CODE                            opcode-eventNotificationCharging
}
opcode-eventNotificationCharging ::= private: 32519
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SSF –> SCF, Timer: Tenc
-- This operation is used by the SSF to report to the SCF the occurence of a specific charging event
-- type as previously requested by the SCF in a RequestNotificationChargingEvent operation.

facilitySelectedAndAvailable    OPERATION ::= {
ARGUMENT                        FacilitySelectedAndAvailableArg
RETURN RESULT                   FALSE
ERRORS                          {applicationError |
                                FailureReport
                                }
CODE                            opcode-facilitySelectedAndAvailable
}
opcode-facilitySelectedAndAvailable ::= private: 25613
-- Family Name is "Request/Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate that a Facility_Selected_and_Available event was detected.
-- (DP Facility_Selected_And_Available).

forwardCall             OPERATION ::= {
ARGUMENT                ForwardCallArg
RETURN RESULT           FALSE
```

```
ERRORS                {applicationError
                      }
CODE                  opcode-forwardCall
}
opcode-forwardCall ::= private: 27137
-- Family Name is "Connectivity Control"
-- Direction: SCF -> SSF.
-- This operation is used to forward a call.

furnishChargingInformation    OPERATION ::= {
ARGUMENT                      FurnishChargingInformationArg
RETURN RESULT                 FALSE
ERRORS                        {MissingParameter|
                              TaskRefused|
                              UnexpectedComponentSequence|
                              UnexpectedDataValue|
                              UnexpectedParameter
                              }
CODE                          opcode-furnishChargingInformation
}
opcode-furnishChargingInformation ::= private: 32520
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF –> SSF, Timer: Tfci
-- This operation is used to request the SSF to generate, register a call record or to include some
-- information in the default call record.The registered call record is intended for off line charging of the call.

instructionsToSRF             OPERATION ::= {
ARGUMENT                      InstructionsToSRFArg
RETURN RESULT                 FALSE
ERRORS                        {applicationError |
                              FailureReport
                              }
CODE                          opcode-instructionsToSRF
}
opcode-instructionsToSRF ::= private: 32528
-- Family Name is "T1 IN ITU operations"
-- Direction: SCF -> SRF, Timer: Tits.
-- This operation is used to request the SRF to perform certain functions, e.g., play announcement and collect digits.

monitorForChange              OPERATION ::= {
ARGUMENT                      MonitorForChangeArg
RETURN RESULT                 FALSE
ERRORS                        {applicationError |
                              FailureReport
                              }
CODE                          opcode-monitorForChange
}
opcode-monitorForChange ::= private:  26369
-- Family Name is "Status Notification"
-- Direction: SCF -> SSF.
-- This operation is used to request that a resource be monitored for changes in its busy/idle status until the
-- desired status is reached or until the monitor duration expires.
```

```
monitorSuccess          OPERATION ::= {
ARGUMENT                MonitorSuccessArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-monitorSuccess
}
opcode-monitorSuccess ::= private:  26371
-- Family Name is "Status Notification"
-- Direction: SSF -> SCF.
-- This operation is sent in response to MonitorForChange, if the facility is not in the designated
-- busy/idle status and the monitorDuration is not zero.

nCAData                 OPERATION ::= {
ARGUMENT                NCADataArg
RETURN RESULT           FALSE
CODE                    opcode-nCAData
}
opcode-nCAData ::= private:  27394
-- Family Name is "NCA Signaling"
-- Direction: SSF -> SCF, SCF -> SSF.
-- This operation is used to send data from the SCF to the ISDN-connected device or from the
-- ISDN-connected device to the SCF.

nCARequest      OPERATION ::- {
ARGUMENT        NCARequestArg
RESULT          SEQUENCE{
                EnvelopeEncodingAuthority   EnvelopeEncodingAuthority,
                envelopContent              EnvelopContent,
                securityEnvelope            SecurityEnvelope        OPTIONAL,
                amp1                        Amp1                    OPTIONAL,
                amp2                        Amp2                    OPTIONAL,
                extensionParameter          ExtensionParameter      OPTIONAL
                }
ERRORS          {applicationError |
                FailureReport
                }
CODE            opcode-nCARequest
}
opcode-nCARequest ::= private:  27393
-- Family Name is "NCA Signaling"
-- Direction: SSF -> SCF, SCF -> SSF.
-- This operation is used to initiate communication between an ISDN-connected device
-- and an SCF.

oAnswer                 OPERATION ::= {
ARGUMENT                OAnswerArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-oAnswer
```

}

opcode-oAnswer ::= private: 25611

-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer: T_{oa}
-- This operation is used for indication from the terminating half BCSM that the call is accepted and
-- answered by terminating party (e.g. terminating party goes offhook, T1.607 Connect message
-- received, ISUP Answer message received) (DP O_Answer).

```
oCalledPartyBusy        OPERATION ::= {
ARGUMENT                OCalledPartyBusyArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-oCalledPartyBusy
}
opcode-oCalledPartyBusy ::= private: 25607
```

-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer: T_{ob}
-- This operation is used for Indication from the terminating half BCSM that the terminating party is busy
-- (DP O_Called_Party_Busy).

```
oDisconnect             OPERATION ::= {
ARGUMENT                ODisconnectArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-oDisconnect
}
opcode-oDisconnect ::= private: 25626
```

-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer: T_{od}
-- This operation is used for a disconnect indication (e.g. onhook, T1.607 Disconnect message,
-- SS7 Release message) is received from the originating party, or received from the terminating party
-- via the terminating half BCSM. (DP O_Disconnect).

```
oDTMFEntered            OPERATION ::= {
ARGUMENT                ODTMFEnteredArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-oDTMFEntered
}
opcode-oDTMFEntered ::= private: 25627
```

-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF.
-- This operation is used for Indication from the originating half BCSM that the user has entered certain DTMF digits.
-- (DP O_Mid_Call).

```
offerCall               OPERATION ::= {
ARGUMENT                OfferCallArg
```

```
RETURN RESULT        FALSE
ERRORS               {applicationError
                     }
CODE                 opcode-offerCall
}
opcode-offerCall ::= private: 25860
-- Family Name is "Connection Control (Call Model)"
-- Direction: SCF -> SSF.
-- This operation is used to continue call processing after triggering on T_Busy.

oMidCall             OPERATION ::= {
ARGUMENT             MidCallArg
RETURN RESULT        FALSE
ERRORS               {MissingCustomerRecord|
                      MissingParameter|
                      ParameterOutOfRange|
                      SystemFailure|
                      TaskRefused|
                      UnexpectedComponentSequence|
                      UnexpectedDataValue|
                      UnexpectedParameter
                     }
CODE                 opcode-oMidCall
}
opcode-oMidCall ::= private: 32522
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SSF –> SCF, Timer: T_omc
-- This operation is used to indicate a feature request is received from the originating party
-- (e.g. hook flash, ISDN feature activation, T1.607 HOLD or RETrieve message).
-- (DP O_Mid_Call).

oNoAnswer            OPERATION ::= {
ARGUMENT             ONoAnswerArg
RETURN RESULT        FALSE
ERRORS               {applicationError |
                     FailureReport
                     }
CODE                 opcode-oNoAnswer
}
opcode-oNoAnswer ::= private: 25609
-- Family Name is "Request/Notification"
-- Direction: SSF → SCF, Timer: Tona
-- This operation is used for indication from the terminating half BCSM that the terminating party does not
-- answer within a specified time period (DP O_No_Answer).

originationAttempt   OPERATION ::= {
ARGUMENT             OriginationAttemptArg
RETURN RESULT        FALSE
ERRORS               {applicationError |
                     FailureReport
                     }
CODE                 opcode-originationAttempt
}
```

```
opcode-originationAttempt ::= private: 25624
-- Family Name is "Request/Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate that an Origination_Attempt trigger/event was detected.
-- (DP Origination_Attempt)

oSuspended              OPERATION ::= {
ARGUMENT                OSuspendedArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-oSuspended
}
opcode-oSuspended ::= private: 25625
-- Family Name is "Request/Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate that an O_Suspended event was detected.
-- (DP O_Suspended)

oTermSeized             OPERATION ::= {
ARGUMENT                OTermSeizedArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-oTermSeized
}
opcode-oTermSeized ::= private: 25612
-- Family Name is "Request/Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate that an O_Term_Seized event was detected.
-- (DP O_Term_Seized)

releaseCall             OPERATION ::= {
ARGUMENT                ReleaseCallArg
RETURN RESULT           FALSE
ERRORS                  {applicationError
                        }
CODE                    opcode-releaseCall
}
opcode-releaseCall ::= private: 25859
-- Family Name is "Connection Control (Call Model)"
-- Direction: SCF –> SSF, Timer: Trc
-- This operation is used to tear down an existing call at any phase of the call for all parties
-- involved in the call.

reportError             OPERATION ::= {
ARGUMENT                ReportErrorArg
RETURN RESULT           FALSE
CODE                    opcode-reportError
}
opcode-reportError ::= private:  28417
```

```
-- Family Name is "Abnormal"
-- Direction: SSF -> SCF.
-- This operation is used to report an application error.

requestNotificationChargingEvent     OPERATION ::= {
ARGUMENT                             RequestNotificationChargingEventArg
RETURN RESULT                        FALSE
ERRORS                               {MissingParameter|
                                     ParameterOutOfRange|
                                     SystemFailure|
                                     TaskRefused|
                                     UnexpectedComponentSequence|
                                     UnexpectedDataValue|
                                     UnexpectedParameter
                                     }
CODE                                 opcode-requestNotificationChargingEvent
}
opcode-requestNotificationChargingEvent ::= private: 32523
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF –> SSF, Timer: Trnc
-- This operation is used by the SCF to instruct the SSF on how to manage the charging events
-- which are received from other FE's and not under control of the service logic instance.

requestReportBCMEvent         OPERATION ::= {
ARGUMENT                      RequestReportBCMEventArg
RETURN RESULT                 FALSE
ERRORS                        {applicationError
                              }
CODE                          opcode-requestReportBCMEvent
}
opcode-requestReportBCMEvent ::= private: 27905
-- Family Name is " Request Event"
-- Direction: SCF -> SSF.
-- This operation is used to request the SSF to monitor for call-related events (e.g., BCM events such as
-- busy or no answer) and send a notification to the SCF if an event occurs.

resetTimer              OPERATION
ARGUMENT                ResetTimerArg
RETURN RESULT           FALSE
ERRORS                  {MissingParameter|
                        ParameterOutOfRange|
                        TaskRefused|
                        UnexpectedComponentSequence|
                        UnexpectedDataValue|
                        UnexpectedParameter
                        }
CODE                    opcode-resetTimer
}
opcode-resetTimer ::= private: 32524
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF –> SSF, Timer: Trt
-- This operation is used to request the SSF to refresh an application timer in the SSF.
```

```
resourceClear         OPERATION ::= {
ARGUMENT              ResourceClearArg
RETURN RESULT         FALSE
ERRORS                {applicationError |
                      FailureReport
                      }
CODE                  opcode-resourceClear
}
opcode-resourceClear ::= private:  26114
-- Family Name is "Caller Interaction"
-- Direction: SSF -> SCF or SRF -> SCF.
-- This operation is used to indicate the result of a request to route a call to a resource.

routeSelectFailure    OPERATION ::= {
ARGUMENT              RouteSelectFailureArg
RETURN RESULT         FALSE
ERRORS                {applicationError |
                      FailureReport
                      }
CODE                  opcode-routeSelectFailure
}
opcode-routeSelectFailure ::= private: 25623
-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer:  Trsf
-- This operation is used to indicate that the SSP is unable to select a route (e.g. unable to determine a
-- correct route, no more routes on route list) or indication from the terminating half BCSM that a call
-- cannot be presented to the terminating party (e.g. network congestion) (DP Route_Select_Failure).

sendChargingInformation       OPERATION
ARGUMENT                      SendChargingInformationArg
RETURN RESULT                 FALSE
ERRORS                        {MissingParameter|
                              UnexpectedComponentSequence|
                              UnexpectedParameter|
                              ParameterOutOfRange|
                              SystemFailure|
                              TaskRefused|
                              UnknownLegID
                              }
CODE                          opcode-sendChargingInformation
}
opcode-sendChargingInformation ::= private: 32525
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SCF –> SSF, Timer: Tsci
-- This operation is used to instruct the SSF on the charging information to send by the SSF. The charging
-- information can either be sent back by means of signalling or internal if the SSF is located in the local
-- exchange. In the local exchange this information may be used to update the charge meter or to create a
-- standard call record.

sendNotification      OPERATION ::= {
ARGUMENT              SendNotificationArg
RETURN RESULT         FALSE
ERRORS                {applicationError
```

```
                          }
CODE                      opcode-sendNotification
}
opcode-sendNotification ::= private:  26373
-- Family Name is "Status Notification"
-- Direction: SCF -> SSF.
-- This operation is used to request notification of call termination.

sendToResource      OPERATION ::= {
ARGUMENT            SendToResourceArg
RESULT              SEQUENCE {
                    extensionParameter         ExtensionParameter             OPTIONAL
                    …
                    }
ERRORS              {applicationError
                    }
CODE                opcode-sendToResource
}
opcode-sendToResource ::= private: 26113
-- Family Name is "Caller Interaction"
-- Direction: SCF -> SSF.
-- This operation is used to request that a call be routed to a resource.

serviceFilteringResponse        OPERATION ::= {
ARGUMENT                        ServiceFilteringResponseArg
RETURN RESULT                   FALSE
CODE                            opcode-serviceFilteringResponse
}
opcode-serviceFilteringResponse ::= private: 32526
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SSF –> SCF, Timer: Tsfr
-- This operation is used to send back to the SCF the values of counters specified in a previous
-- ActivateServiceFiltering operation.

statusReported            OPERATION ::= {
ARGUMENT                  StatusReportedArg
RETURN RESULT             FALSE
ERRORS                    {applicationError
                          }
CODE                      opcode-statusReported
}
opcode-statusReported ::= private:  26370
-- Family Name is "Status Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate the busy/idle status of a resource.

tAnswer                   OPERATION ::= {
ARGUMENT                  TAnswerArg
RETURN RESULT             FALSE
ERRORS                    {applicationError |
                          FailureReport
                          }
CODE                      opcode-tAnswer
```

```
}
opcode-tAnswer ::= private: 25610
-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer: Tta
-- This operation is used to indicate that the call is accepted and answered by terminating party
-- (e.g. terminating party goes offhook, T1.607 Connect message received, ISUP Answer
-- message received) (DP T_Answer).

tBusy                   OPERATION ::= {
ARGUMENT                TBusyArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-tBusy
}
opcode-tBusy ::= private: 25606
-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer: Ttb
-- This operation is used to indicate all resources in group busy (DP TBusy).

terminationAttempt      OPERATION ::= {
ARGUMENT                TerminationAttemptArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-terminationAttempt
}
opcode- terminationAttempt ::= private: 25605
-- Family Name is "Request/Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate that a Termination_Attempt trigger was detected.
-- (DP Termination_Attempt)

terminationNotification      OPERATION ::= {
ARGUMENT                     TerminationNotificationArg
RETURN RESULT                FALSE
CODE                         opcode-terminationNotification
}
opcode-terminationNotification ::= private:  26372
-- Family Name is "Status Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate notification of call termination.

timeout                 OPERATION ::= {
ARGUMENT                TimeoutArg
RETURN RESULT           FALSE
ERRORS                  {applicationError |
                        FailureReport
                        }
CODE                    opcode-timeout
}
```

```
opcode-timeout ::= private:  25620
-- Family Name is "Request/Notification"
-- Direction: SSF -> SCF.
-- This operation is used to indicate that a timeout event was detected.

tMidCall            OPERATION ::= {
ARGUMENT            MidCallArg
RETURN RESULT       FALSE
ERRORS              {MissingCustomerRecord|
                    MissingParameter|
                    ParameterOutOfRange|
                    SystemFailure|
                    TaskRefused|
                    UnexpectedComponentSequence|
                    UnexpectedDataValue|
                    UnexpectedParameter
                    }
CODE                opcode-tMidCall
}
opcode-tMidCall ::= private: 32527
-- Family Name is "T1 IN phase 1 ITU operations"
-- Direction: SSF –> SCF, Timer: Ttmc
-- This operation is used to indicate that a feature request is received from the terminating party (e.g. hook
-- flash, ISDN feature activation T1.607 HOLD or RETrieve message). (DP T_Mid_Call).

tNoAnswer           OPERATION ::= {
ARGUMENT            TNoAnswerArg
RETURN RESULT       FALSE
ERRORS              {applicationError |
                    FailureReport
                    }
CODE                opcode-tNoAnswer
}
opcode-tNoAnswer ::= private: 25608
-- Family Name is "Request/Notification"
-- Direction: SSF –> SCF, Timer: Ttna
-- This operation is used to indicate that the terminating party does not answer within a specified duration.
-- (DP T_No_Answer).

update              OPERATION ::= {
ARGUMENT            UpdateArg
ERRORS              {applicationError|
                    FailureReport
                    }
CODE                opcode-update
}
RESULT SEQUENCE {  amp1                   Amp1                  OPTIONAL,
                   amp2                   Amp2                  OPTIONAL,
                   extensionParameter     ExtensionParameter    OPTIONAL
}
opcode-update ::= private:  26627
-- Family Name is "Information Revision"
-- Direction: SCF –> SSF
```

```
-- This operation is used by the SCF to request that certain SSF information (e.g., a trigger.
-- activation/deactivation, O_No_Answer and T_No_Answer timers) be changed.

updateRequest           OPERATION ::= {
ARGUMENT                UpdateRequestArg
ERRORS                  {failureCause
                        }
CODE                    opcode-updateRequest
}
RESULT SEQUENCE {       failureCause        FailureCause                              OPTIONAL
}
opcode-updateRequest ::= private:  26625
-- Family Name is "Information Revision"
-- Direction: SCF -> SSF
-- This operation is used by the SCF to request that certain SSF information (e.g., a trigger.
-- activation/deactivation) be changed.

END
```

The following value ranges do apply for operation-specific timers in INAP:

- short: 1 - 10 seconds
- *medium*: 1 - 60 seconds
- long: 1 second - 30 minutes

Table 11 lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network-specific and has to be defined by the network operator.

Table 11 - Operation timer values

Operation Name	*Timer*	*value range*
ActivateServiceFiltering	T_{asf}	*medium*
ActivityTest	T_{at}	*short*
ApplyCharging	T_{ac}	*short*
ApplyChargingReport	T_{acr}	*short*
AssistRequestInstructions	T_{ari}	*short*
CallInformationReport	T_{cirp}	*short*
CallInformationRequest	T_{cirq}	*short*
EventNotificationCharging	T_{enc}	*short*
FurnishChargingInformation	T_{fci}	*short*
InstructionsToSRF	T_{its}	*short*
OmidCall	T_{omc}	*short*
ReleaseCall	T_{rc}	*short*
RequestNotificationChargingEvent	T_{rnc}	*short*
ResetTimer	T_{rt}	*short*
SendChargingInformation	T_{sci}	*short*
ServiceFilteringResponse	T_{sfr}	*short*
TmidCall	T_{tmc}	*short*

7.3.1.2 T1 IN error types

ANSI-IN-Errors {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inaperrors(1) version1(0)}

-- This module contains the type definitions for theANSI-IN errors.
-- Where a parameter of type CHOICE is tagged with a specific tag value, the tag is automatically
-- replaced with an EXPLICIT tag of the same value.

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- exports everything

IMPORTS

ERROR

FROM IN-Remote-Operations-Information-Objects {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) informationObjects(5) version1(0)}

InvokeID,

UnavailableNetworkResource

FROM ANSI-IN-DataTypes {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inapdatatypes(2) version1(0)}

```
-- TYPE DEFINITION FOR  T1 IN ERRORS FOLLOWS

applicationError        ERROR
PARAMETER               SEQUENCE{
                        applicationErrorString      ApplicationErrorString,
                        extensionParameter          ExtensionParameter          OPTIONAL
                        …
                        }
::= private: 1
-- This message is used to report an application error in an existing TCAP transaction.

failureReport                       ERROR
PARAMETER                           SEQUENCE {
            failureCause            FailureCause,
            failureCauseData        FailureCauseData                OPTIONAL,
            extensionParameter      ExtensionParameter              OPTIONAL
            …
            }
::= private: 2
-- This message is used to indicate that a failure was detected.

ImproperCallerResponse ::= ERROR
-- The caller response was not as expected.

MissingCustomerRecord ::= ERROR
-- The Service Logic Program could not be found in the SCF.

MissingParameter ::= ERROR
-- An expected optional parameter was not received.

ParameterOutOfRange ::= ERROR
-- The parameter was not as expected (e.g. missing or out of range).

RequestedInfoError ::= ERROR
            PARAMETER ENUMERATED {
            unknownRequestedInfo(1),
            requestedInfoNotAvailable(2)
            }
-- The requested information cannot be found.

SystemFailure ::= ERROR
            PARAMETER {
            unavailableNetworkResource      UnavailableNetworkResource
            }
-- The operation could not be completed due to a system failure at the serving physical entity.

TaskRefused ::= ERROR
            PARAMETER ENUMERATED {
            generic(0),
            unobtainable (1),
            congestion(2)
```

}

-- An entity normally capable of the task requested cannot or chooses not to perform the task at this
-- time. This includes error situations such as congestion and unobtainable address as used in e.g. the
-- connect operation.)

UnavailableResource ::= ERROR
-- A requested resource is not available at the serving entity.

UnexpectedComponentSequence ::= ERROR
-- An incorrect sequence of Components was received.

UnexpectedDataValue ::= ERROR
-- The data value was not as expected (e.g. routing number expected but billing number received)

UnexpectedParameter ::= ERROR
-- A parameter received was not expected.

UnknownLegID ::= ERROR
-- Leg not known to the SSF.

UnknownResource ::= ERROR
-- Resource whose status is being requested is not known to the serving entity.

END

7.3.1.3 T1 IN data types

ANSI-IN-Datatypes {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inapdatatypes(2) version1(0)}

-- This module contains the type definitions for the T1 IN data types.
-- Where a parameter of type CHOICE is tagged with a specific tag value, the tag is automatically
-- replaced with an EXPLICIT tag of the same value
-- The following parameters map onto bearer protocol (i.e., T1.607 and ISUP) parameters:
-- CallingPartySubaddress, CalledPartyNumber,
-- Prefix (derived from dialed digits), DestinationRoutingAddress,
-- DialedDigits, ISDNAccessRelatedInformation, CallingPartysCategory, LocationNumber,
-- TravellingClassMark, AlertingPattern (T1.607 only),
-- ReleaseCause (and other Cause parameters), ServiceProfileIdentifier (T1.610 only),
-- BearerCapability, CallingPartyNumber, HighLayerCompatibility, OriginalCalledPartyID,
-- RedirectingPartyID, and RedirectionInformation.
-- The following SSF parameters do not map onto bearer protocol (i.e., T1.607 and ISUP)
-- parameters and therefore are assumed to be local to the switching system: CallingPartyBusinessGroupID
-- FacilityGroup, FacilityGroupMember, RouteList, LegID,
-- ForwardingCondition, CorrelationID, TriggerType and ServiceKey.
-- Where possible, Network Operators should specify the maximum size within their network of
-- parameters specified in this Standard that are of an indeterminate length.

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- exports everything

IMPORTS

```
                            InvokeIdType
FROM IN-Remote-Operations-Information-Objects {iso(1) memberbody(2) us(840) t1-667-2002(10069)
modules(0) informationObjects(5) version1(0)};
-- TYPE DEFINITIONS FOR T1 IN DATA TYPES FOLLOWS
-- Argument data types
-- The ordering of parameters in the argument sequences has been arbitrary. Further study may be
-- required to order arguments in a manner which will facilitate efficient encoding and decoding.

AcgArg ::= SEQUENCE{
        controlType                    ControlCauseIndicator,
        gapIndicatorsDuration          GapDuration,
        gapIndicatorsGapInterval       GapInterval,
        translationType                TranslationType,
        globalTitleAddress             GlobalTitleAddress,
        extensionParameter             ExtensionParameter                  OPTIONAL
        …
        }

ActivateServiceFilteringArg ::= SEQUENCE {
        filteredCallTreatment          [0] FilteredCallTreatment,
        filteringCharacteristics       [1] FilteringCharacteristics,
        filteringTimeOut               [2] FilteringTimeOut,
        filteringCriteria              [3] FilteringCriteria,
        startTime                      [4] DateAndTime                     OPTIONAL,
        extensions                     [5] SEQUENCE SIZE(1..numOfExtensions)
OF
                                       ExtensionParameter                  OPTIONAL
        …
        }

AnalyzedInformationArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey UserID,
        bearerCapability                         BearerCapability,
        calledPartyNumber                        CalledPartyID             OPTIONAL,
        servingAreaID                            Lata                      OPTIONAL,
        serviceAddressInformationTriggerType     TriggerCriteriaType       OPTIONAL,
        chargeNumber                             ChargeNumber              OPTIONAL,
        callingPartyNumber                       CallingPartyID            OPTIONAL,
        callingPartyBusinessGroupID              CallingPartyBGID          OPTIONAL,
        callingPartysCategory                    ChargePartyStationType    OPTIONAL,
        carrier                                  Carrier                   OPTIONAL,
        accessCode                               AccessCode                OPTIONAL,
        dialledDigitsCAI                         CollectedAddressInfo      OPTIONAL,
        dialledDigitsCD                          CollectedDigits           OPTIONAL,
        featureCode                              VerticalServiceCode       OPTIONAL,
        travellingClassMark                      Tcm                       OPTIONAL,
        originalCalledPartyID                    OriginalCalledPartyID     OPTIONAL,
        redirectingPartyID                       RedirectingPartyID        OPTIONAL,
        redirectionInformation                   RedirectionInformation    OPTIONAL,
        cGEncountered                            ACGEncountered            OPTIONAL,
        amp1                                     Amp1                      OPTIONAL,
        amp2                                     Amp2                      OPTIONAL,
        sap                                      Sap                       OPTIONAL,
```

```
        sTRConnection                   STRConnection                   OPTIONAL,
        aMASequenceNumber               AMASequenceNumber               OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL,
        genericAddressList              GenericAddressList              OPTIONAL,
        networkSpecificFacilities       NetworkSpecificFacilities       OPTIONAL,
        forwardCallIndicator            ForwardCallIndicator            OPTIONAL,
        …
        genericDigitsList               GenericDigitsList               OPTIONAL,
        callingGeodeticLocation         CallingGeodeticLocation         OPTIONAL
        …
        }

-- For the OPTIONAL parameters, see 7.4.6 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

AnalyzeRouteArg ::= SEQUENCE {
        chargeNumber                    ChargeNumber                    OPTIONAL,
        callingPartyNumber              CallingPartyID                  OPTIONAL,
        callingPartysCategory           ChargePartyStationType          OPTIONAL,
        calledPartyNumber               CalledPartyID                   OPTIONAL,
        destinationRoutingAddressON     OutpulseNumber                  OPTIONAL,
        travellingClassMark             Tcm                             OPTIONAL,
        routeListPTG                    PrimaryTrunkGroup               OPTIONAL,
        routeListATG                    AlternateTrunkGroup             OPTIONAL,
        routeListSATG                   SecondAlternateTrunkGroup       OPTIONAL,
        carrier                         Carrier                         OPTIONAL,
        carrierAC                       AlternateCarrier                OPTIONAL,
        carrierSAC                      SecondAlternateCarrier          OPTIONAL,
        alertingPattern                 PassiveLegTreatment             OPTIONAL,
        redirectingPartyID              RedirectingPartyID              OPTIONAL,
        primaryBillingIndicator         PrimaryBillingIndicator         OPTIONAL,
        alternateBillingIndicator       AlternateBillingIndicator       OPTIONAL,
        secondAlternateBillingIndicator SecondAlternateBillingIndicator OPTIONAL,
        overflowBillingIndicator        OverflowBillingIndicator        OPTIONAL,
        aMAAlternateBillingNumber       AMAAlternateBillingNumber       OPTIONAL,
        aMABusinessCustomerID           AMABusinessCustomerID           OPTIONAL,
        aMALineNumber                   SEQUENCE SIZE (1..2) OF AMALineNumber   OPTIONAL,
        aMAslpID                        AMAslpID                        OPTIONAL,
        aMADigitsDialedWC               SEQUENCE SIZE (1..2) OF AMADigitsDialedWC OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        serviceProviderID               ServiceProviderID               OPTIONAL,
        serviceContext                  ServiceContext                  OPTIONAL,
        aMABillingFeature               AMABillingFeature               OPTIONAL,
        aMASequenceNumber               AMASequenceNumber               OPTIONAL,
        redirectionInformation          RedirectionInformation          OPTIONAL,
        carrierUsage                    CarrierUsage                    OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL,
        genericAddressList              GenericAddressList              OPTIONAL,
        networkSpecificFacilities       NetworkSpecificFacilities       OPTIONAL,
        callingPartyBGID                CallingPartyBGID                OPTIONAL,
        forwardCallIndicator            ForwardCallIndicator            OPTIONAL,
```

```
        aMAserviceProviderID          AMAServiceProviderID                        OPTIONAL,
        …
        destinationRoutingAddress     DestinationRoutingAddress                   OPTIONAL,
        destinationRoutingAddressUsage       DestinationRoutingAddressUsage       OPTIONAL
        …
        }

ApplyChargingArg ::= SEQUENCE {
        aChBillingChargingCharacteristics  [0] AChBillingChargingCharacteristics,
        partyToCharge                      [2] LegID                              OPTIONAL,
        extensions                         [3] SEQUENCE SIZE(1..numOfExtensions)  OF
                                           ExtensionParameter                     OPTIONAL
        …
        }

-- The partyToCharge parameter indicates the party in the call to which the ApplyCharging operation should
-- be applied. If this parameter is not present, then the ApplyCharging operation is applied to the A-party

ApplyChargingReportArg ::= CallResult

AssistRequestInstructionsArg ::= SEQUENCE {
        sCFCorrelationID              EchoData,
        extensionParameter            ExtensionParameter                          OPTIONAL
        …
        }

-- OPTIONAL denotes network operator specific use.

AuthorizeTerminationArg  ::= SEQUENCE {
        callingPartyNumber            CallingPartyID                              OPTIONAL,
        travellingClassMark           Tcm                                         OPTIONAL,
        alertingPattern               ControllingLegTreatment                     OPTIONAL,
        displayInformation            DisplayText                                 OPTIONAL,
        primaryBillingIndicator       PrimaryBillingIndicator                     OPTIONAL,
        aMAAlternateBillingNumber     AMAAlternateBillingNumber                   OPTIONAL,
        aMABusinessCustomerID         AMABusinessCustomerID                       OPTIONAL,
        aMALineNumber                 SEQUENCE SIZE (1..2) OF AMALineNumber       OPTIONAL,
        aMAslpID                      AMAslpID                                    OPTIONAL,
        aMADigitsDialedWC             SEQUENCE SIZE (1..2) OF AMADigitsDialedWC   OPTIONAL,
        amp1                          Amp1                                        OPTIONAL,
        amp2                          Amp2                                        OPTIONAL,
        serviceProviderID             ServiceProviderID                           OPTIONAL,
        serviceContext                ServiceContext                              OPTIONAL,
        aMABillingFeature             AMABillingFeature                           OPTIONAL,
        aMASequenceNumber             AMASequenceNumber                           OPTIONAL,
        extensionParameter            ExtensionParameter                          OPTIONAL,
        aMAserviceProviderID          AMAServiceProviderID                        OPTIONAL
        …
        }

CallInfoFromResourceArg ::= SEQUENCE{
        iPReturnBlock                 IPReturnBlock                               OPTIONAL,
```

```
        amp1                        Amp1                                    OPTIONAL,
        amp2                        Amp2                                    OPTIONAL,
        extensionParameter          ExtensionParameter                      OPTIONAL
        …
        }

CallInfoFromResourceRR SEQUENCE{
        resourceType                ResourceType                            OPTIONAL,
        strParameterBlock           StrParameterBlock                       OPTIONAL,
        amp1                        Amp1                                    OPTIONAL,
        amp2                        Amp2                                    OPTIONAL,
        serviceProviderID           ServiceProviderID                       OPTIONAL,
        serviceContext              ServiceContext                          OPTIONAL,
        extensionParameter          ExtensionParameter                      OPTIONAL,
        disconnectFromIPForbidden   BOOLEAN DEFAULT TRUE
        …
        }

CallInformationReportArg ::= SEQUENCE {
        requestedInformationList    [0] RequestedInformationList,
        correlationID               [1] CorrelationID                       OPTIONAL,
        extensions                  [2] SEQUENCE SIZE(1..numOfExtensions)   OF
                                    ExtensionParameter                      OPTIONAL
        …
        }

-- OPTIONAL denotes network operator optional.

CallInformationRequestArg ::= SEQUENCE {
        requestedInformationTypeList [0] RequestedInformationTypeList,
        correlationID               [1] CorrelationID                       OPTIONAL,
        extensions                  [2] SEQUENCE SIZE(1..numOfExtensions)   OF
                                    ExtensionParameter                      OPTIONAL
        …
        }

-- OPTIONAL denotes network operator optional.

CancelResourceEventArg  ::= SEQUENCE{
        amp1                        Amp1                                    OPTIONAL,
        amp2                        Amp2                                    OPTIONAL,
        serviceProviderID           ServiceProviderID                       OPTIONAL,
        serviceContext              ServiceContext                          OPTIONAL,
        extensionParameter          ExtensionParameter                      OPTIONAL
        …
        }

CancelStatusReportRequestArg ::= SEQUENCE {
        resourceID                  [0] ResourceID                          OPTIONAL,
        extensions                  [1] SEQUENCE SIZE(1..numOfExtensions)   OF
                                    [1] ExtensionParameter                  OPTIONAL
        …
```

```
        }

CloseArg ::=  SEQUENCE{
        serviceAddressInformationServiceKey  UserID                 OPTIONAL,
        bearerCapability                     BearerCapability       OPTIONAL,
        closeCause                           CloseCause             OPTIONAL,
        amp1                                 Amp1                   OPTIONAL,
        amp2                                 Amp2                   OPTIONAL,
        extensionParameter                   ExtensionParameter     OPTIONAL
        …
        }

CollectedInformationArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey UserID,
        bearerCapability                     BearerCapability,
        chargeNumber                         ChargeNumber           OPTIONAL,
        servingAreaID                        Lata                   OPTIONAL,
        carrier                              Carrier                OPTIONAL,
        serviceAddressInformationTriggerType TriggerCriteriaType    OPTIONAL,
        callingPartyNumber                   CallingPartyID         OPTIONAL,
        callingPartysCategory                ChargePartyStationType OPTIONAL,
        accessCode                           AccessCode             OPTIONAL,
        dialledDigitsCAI                     CollectedAddressInfo   OPTIONAL,
        dialledDigitsCD                      CollectedDigits        OPTIONAL,
        featureCode                          VerticalServiceCode    OPTIONAL,
        travellingClassMark                  Tcm                    OPTIONAL,
        originalCalledPartyID                OriginalCalledPartyID  OPTIONAL,
        redirectingPartyID                   RedirectingPartyID     OPTIONAL,
        redirectionInformation               RedirectionInformation OPTIONAL,
        cGEncountered                        ACGEncountered         OPTIONAL,
        amp1                                 Amp1                   OPTIONAL,
        amp2                                 Amp2                   OPTIONAL,
        sap                                  Sap                    OPTIONAL,
        genericAddressList                   GenericAddressList     OPTIONAL,
        aMASequenceNumber                    AMASequenceNumber      OPTIONAL,
        extensionParameter                   ExtensionParameter     OPTIONAL,
        …
        genericDigitsList                    GenericDigitsList      OPTIONAL,
        callingGeodeticLocation              CallingGeodeticLocation OPTIONAL
        …
        }

-- For the OPTIONAL parameters, See 7.4.6 for the trigger detection point processing rules to specify
-- when these parameters are included in the message.

CollectInformationArg ::= SEQUENCE{
        callingPartyNumber              CallingPartyID                  OPTIONAL,
        dialedDigitsCD                  CollectedDigits                 OPTIONAL,
        dpConverter                     DPConverter                     OPTIONAL,
        primaryBillingIndicator         PrimaryBillingIndicator         OPTIONAL,
        alternateBillingIndicator       AlternateBillingIndicator       OPTIONAL,
        secondAlternateBillingIndicator SecondAlternateBillingIndicator OPTIONAL,
```

```
          overflowBillingIndicator          OverflowBillingIndicator                   OPTIONAL,
          aMAAlternateBillingNumber         AMAAlternateBillingNumber                  OPTIONAL,
          aMALineNumber                     SEQUENCE SIZE (1..2) OF AMALineNumber      OPTIONAL,
          aMAslpID                          AMAslpID                                   OPTIONAL,
          aMADigitsDialedWC                 SEQUENCE SIZE (1..2) OF AMADigitsDialedWC  OPTIONAL,
          amp1                              Amp1                                       OPTIONAL,
          amp2                              Amp2                                       OPTIONAL,
          serviceProviderID                 ServiceProviderID                          OPTIONAL,
          serviceContext                    ServiceContext                             OPTIONAL,
          aMABillingFeature                 AMABillingFeature                          OPTIONAL,
          aMASequenceNumber                 AMASequenceNumber                          OPTIONAL,
          extensionParameter                ExtensionParameter                         OPTIONAL,
          alternateDialingPlanInd           AlternateDialingPlanInd                    OPTIONAL,
          aMAserviceProviderID              AMAServiceProviderID                       OPTIONAL
          …
          }

ContinueArg  ::= SEQUENCE {
          primaryBillingIndicator           PrimaryBillingIndicator                    OPTIONAL,
          aMAAlternateBillingNumber         AMAAlternateBillingNumber                  OPTIONAL,
          aMABusinessCustomerID             AMABusinessCustomerID                      OPTIONAL,
          aMALineNumber                     SEQUENCE SIZE (1..2) OF AMALineNumber      OPTIONAL,
          aMAslpID                          AMAslpID                                   OPTIONAL,
          aMADigitsDialedWC                 SEQUENCE SIZE (1..2) OF AMADigitsDialedWC  OPTIONAL,
          amp1                              Amp1                                       OPTIONAL,
          amp2                              Amp2                                       OPTIONAL,
          serviceProviderID                 ServiceProviderID                          OPTIONAL,
          serviceContext                    ServiceContext                             OPTIONAL,
          aMABillingFeature                 AMABillingFeature                          OPTIONAL,
          aMASequenceNumber                 AMASequenceNumber                          OPTIONAL,
          extensionParameter                ExtensionParameter                         OPTIONAL,
          aMAserviceProviderID              AMAServiceProviderID                       OPTIONAL
          …
          }

EventNotificationChargingArg ::= SEQUENCE {
          eventTypeCharging                 [0] EventTypeCharging,
          eventSpecificInformationCharging  [1] EventSpecificInformationCharging       OPTIONAL,
          legID                             [2] LegID                                  OPTIONAL,
          extensions                        [3] SEQUENCE SIZE(1..numOfExtensions)
                                            OF ExtensionParameter                      OPTIONAL,
          monitorMode                       [30] MonitorMode          DEFAULT notifyAndContinue
          …
          }

-- OPTIONAL denotes network operator-specific use.

FacilitySelectedAndAvailableArg  ::= SEQUENCE {
          serviceAddressInformationServiceKey UserID,
          bearerCapability                  BearerCapability,
          serviceAddressInformationMiscCallInfo NotificationIndicator                  OPTIONAL,
          amp1                              Amp1                                       OPTIONAL,
```

```
        amp2                            Amp2                                  OPTIONAL,
        extensionParameter              ExtensionParameter                    OPTIONAL,
        calledPartyNumber               CalledPartyID                         OPTIONAL,
        servingAreaID                   Lata                                  OPTIONAL,
        serviceAddressInformationTriggerType TriggerCriteriaType              OPTIONAL,
        chargeNumber                         ChargeNumber                     OPTIONAL,
        callingPartyNumber                   CallingPartyID                   OPTIONAL,
        callingPartysCategory                ChargePartyStationType           OPTIONAL,
        originalCalledPartyID                OriginalCalledPartyID            OPTIONAL,
        redirectingPartyID                   RedirectingPartyID               OPTIONAL,
        redirectionInformation               RedirectionInformation           OPTIONAL,
        calledPartysCategory                 CalledPartyStationType           OPTIONAL,
        sap                                  Sap                              OPTIONAL,
        displayInformationGN            GenericName                           OPTIONAL,
        cGEncountered                   ACGEncountered                        OPTIONAL,
        sTRConnection                   STRConnection                         OPTIONAL,
        aMASequenceNumber               AMASequenceNumber                     OPTIONAL
        …
        }

ForwardCallArg  ::= SEQUENCE {
        callingPartyNumber              CallingPartyID                        OPTIONAL,
        chargeNumber                    ChargeNumber                          OPTIONAL,
        callingPartysCategory           ChargePartyStationType                OPTIONAL,
        calledPartyNumber               CalledPartyID                         OPTIONAL,
        destinationRoutingAddressON     OutpulseNumber                        OPTIONAL,
        travellingClassMark             Tcm                                   OPTIONAL,
        routeListPTG                    PrimaryTrunkGroup                     OPTIONAL,
        routeListATG                    AlternateTrunkGroup                   OPTIONAL,
        routeListSATG                   SecondAlternateTrunkGroup             OPTIONAL,
        carrier                         Carrier                               OPTIONAL,
        carrierAC                       AlternateCarrier                      OPTIONAL,
        carrierSAC                      SecondAlternateCarrier                OPTIONAL,
        alertingPattern                 PassiveLegTreatment                   OPTIONAL,
        primaryBillingIndicator         PrimaryBillingIndicator               OPTIONAL,
        alternateBillingIndicator       AlternateBillingIndicator             OPTIONAL,
        secondAlternateBillingIndicator SecondAlternateBillingIndicator       OPTIONAL,
        overflowBillingIndicator        OverflowBillingIndicator              OPTIONAL,
        aMAAlternateBillingNumber       AMAAlternateBillingNumber             OPTIONAL,
        aMABusinessCustomerID           AMABusinessCustomerID                 OPTIONAL,
        aMALineNumber                   SEQUENCE SIZE (1..2) OF AMALineNumber OPTIONAL,
        aMAslpID                        AMAslpID                              OPTIONAL,
        aMADigitsDialedWC               SEQUENCE SIZE (1..2) OF AMADigitsDialedWC OPTIONAL,
        amp1                            Amp1                                  OPTIONAL,
        amp2                            Amp2                                  OPTIONAL,
        serviceProviderID               ServiceProviderID                     OPTIONAL,
        serviceContext                  ServiceContext                        OPTIONAL,
        aMABillingFeature               AMABillingFeature                     OPTIONAL,
        aMASequenceNumber               AMASequenceNumber                     OPTIONAL,
        redirectingPartyID              RedirectingPartyID                    OPTIONAL,
        redirectionInformation          RedirectionInformation                OPTIONAL,
        carrierUsage                    CarrierUsage                          OPTIONAL,
```

```
          extensionParameter           ExtensionParameter                        OPTIONAL,
          aMAserviceProviderID         AMAServiceProviderID                      OPTIONAL,
          …
          destinationRoutingAddress    DestinationRoutingAddress                 OPTIONAL,
          destinationRoutingAddressUsage        DestinationRoutingAddressUsage   OPTIONAL
          …
          }

FurnishChargingInformationArg ::= FCIBillingChargingCharacteristics

InstructionsToSRFArg ::=SEQUENCE{
          resourceType                 ResourceType,
          strParameterBlock            STRParameterBlock,
          disconnectFromIPForbidden    BOOLEAN  DEFAULT TRUE,
          extensionParameter           ExtensionParameter                        OPTIONAL
          …
          }

MonitorForChangeArg ::= SEQUENCE{
          resourceStatus               FacilityStatus,
          monitorDuration              MonitorTime,
          calledPartyNumber            CalledPartyID                             OPTIONAL,
          facilityGID                  FacilityGID                               OPTIONAL,
          resourceIDFGMID              FacilityMemberID                          OPTIONAL,
          bearerCapability             BearerCapability                          OPTIONAL,
          amp1                         Amp1                                      OPTIONAL,
          amp2                         Amp2                                      OPTIONAL,
          extensionParameter           ExtensionParameter                        OPTIONAL
          …
          }

MonitorSuccessArg ::=SEQUENCE{
          resourceStatus               FacilityStatus,
          amp1                         Amp1                                      OPTIONAL,
          amp2                         Amp2                                      OPTIONAL,
          extensionParameter           ExtensionParameter                        OPTIONAL
          …
          }

NCADataArg ::= SEQUENCE{
          CHOICE {
                                       calledPartyNumber                         CalledPartyID
                                       srhrGroupID                               SrhrGroupID},
          callingPartyNumber           CallingPartyID,
          envelopeEncodingAuthority    EnvelopeEncodingAuthority,
          envelopContent               EnvelopContent,
          securityEnvelope             SecurityEnvelope                          OPTIONAL,
          amp1                         Amp1                                      OPTIONAL,
          amp2                         Amp2                                      OPTIONAL,
          cGEncountered                ACGEncountered                            OPTIONAL,
          extensionParameter           ExtensionParameter                        OPTIONAL
          …
```

```
        }

NCARequestArg ::= SEQUENCE{
        calledPartyNumber               CalledPartyID,
        callingPartyNumber              CallingPartyID,
        envelopeEncodingAuthority       EnvelopeEncodingAuthority,
        envelopContent                  EnvelopContent,
        securityEnvelope                SecurityEnvelope                OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        cGEncountered                   ACGEncountered                  OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL
        …
        }

OAnswerArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey   UserID,
        bearerCapability                BearerCapability,
        serviceAddressInformationMiscCallInfo   NotificationIndicator   OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL,
        localSSPTime                    LocalSSPTime                    OPTIONAL
        …
        }

-- For the OPTIONAL parameters, See 7.4.6  for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

OCalledPartyBusyArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey UserID,
        bearerCapability                BearerCapability,
        calledPartyNumber               CalledPartyID                   OPTIONAL,
        servingAreaID                   Lata                            OPTIONAL,
        serviceAddressInformationTriggerType  TriggerCriteriaType       OPTIONAL,
        chargeNumber                    ChargeNumber                    OPTIONAL,
        callingPartyNumber              CallingPartyID                  OPTIONAL,
        callingPartysCategory           ChargePartyStationType          OPTIONAL,
        carrier                         Carrier                         OPTIONAL,
        originalCalledPartyID           OriginalCalledPartyID           OPTIONAL,
        redirectingPartyID              RedirectingPartyID              OPTIONAL,
        redirectionInformation          RedirectionInformation          OPTIONAL,
        busyCause                       BusyCause                       OPTIONAL,
        sap                             Sap                             OPTIONAL,
        serviceAddressInformationMiscCallInfo    NotificationIndicator  OPTIONAL,
        cGEncountered                   ACGEncountered                  OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        sTRConnection                   STRConnection                   OPTIONAL,
        aMASequenceNumber               AMASequenceNumber               OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL
        …
```

```
            }

-- For the OPTIONAL parameters,  See 7.4.6 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

ODisconnectArg  ::= SEQUENCE {
            serviceAddressInformationServiceKey   UserID,
            bearerCapability                       BearerCapability,
            serviceAddressInformationMiscCallInfo    NotificationIndicator        OPTIONAL,
            amp1                                   Amp1                          OPTIONAL,
            amp2                                   Amp2                          OPTIONAL,
            extensionParameter                     ExtensionParameter            OPTIONAL,
            disconnectCause                        DisconnectCause               OPTIONAL,
            localSSPTime                           LocalSSPTime                  OPTIONAL
            …
            }

-- For the OPTIONAL parameters, See 7.4.6  for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

ODTMFEnteredArg  ::= SEQUENCE {
            serviceAddressInformationServiceKey   UserID,
            bearerCapability                       BearerCapability,
            serviceAddressInformationMiscCallInfo    NotificationIndicator        OPTIONAL,
            dTMFDigitsDetected                     DTMFDigitsDetected            OPTIONAL,
            amp1                                   Amp1                          OPTIONAL,
            amp2                                   Amp2                          OPTIONAL,
            extensionParameter                     ExtensionParameter            OPTIONAL,
            localSSPTime                           LocalSSPTime                  OPTIONAL
            …
            }

-- For the OPTIONAL parameters, See 7.4.6  for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

OfferCallArg  ::= SEQUENCE {
            callingPartyNumber          CallingPartyID                                OPTIONAL,
            alertingPattern             ControllingLegTreatment                       OPTIONAL,
            displayInformation          DisplayText                                   OPTIONAL,
            primaryBillingIndicator     PrimaryBillingIndicator                       OPTIONAL,
            aMAAlternateBillingNumber   AMAAlternateBillingNumber                     OPTIONAL,
            aMABusinessCustomerID       AMABusinessCustomerID                         OPTIONAL,
            aMALineNumber               SEQUENCE SIZE (1..2) OF AMALineNumber         OPTIONAL,
            aMAslpID                    AMAslpID                                      OPTIONAL,
            aMADigitsDialedWC           SEQUENCE SIZE (1..2) OF AMADigitsDialedWC     OPTIONAL,
            amp1                        Amp1                                          OPTIONAL,
            amp2                        Amp2                                          OPTIONAL,
            serviceProviderID           ServiceProviderID                             OPTIONAL,
            serviceContext              ServiceContext                                OPTIONAL,
            aMABillingFeature           AMABillingFeature                             OPTIONAL,
            aMASequenceNumber           AMASequenceNumber                             OPTIONAL,
            extensionParameter          ExtensionParameter                            OPTIONAL,
            aMAserviceProviderID        AMAServiceProviderID                          OPTIONAL
```

```
        …
        }

ONoAnswerArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey UserID,
        bearerCapability                BearerCapability,
        calledPartyNumber               CalledPartyID                   OPTIONAL,
        servingAreaID                   Lata                            OPTIONAL,
        serviceAddressInformationTriggerType    TriggerCriteriaType     OPTIONAL,
        chargeNumber                    ChargeNumber                    OPTIONAL,
        callingPartyNumber              CallingPartyID                  OPTIONAL,
        callingPartysCategory           ChargePartyStationType          OPTIONAL,
        carrier                         Carrier                         OPTIONAL,
        originalCalledPartyID           OriginalCalledPartyID           OPTIONAL,
        redirectingPartyID              RedirectingPartyID              OPTIONAL,
        redirectionInformation          RedirectionInformation          OPTIONAL,
        sap                             Sap                             OPTIONAL,
        serviceAddressInformationMiscCallInfo   NotificationIndicator   OPTIONAL,
        cGEncountered                   ACGEncountered                  OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        sTRConnection                   STRConnection                   OPTIONAL,
        aMASequenceNumber               AMASequenceNumber               OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL
        …
        }

-- For the OPTIONAL parameters, See 7.4.6  for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

OriginationAttemptArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey     UserID,
        bearerCapability                BearerCapability,
        chargeNumber                    ChargeNumber                    OPTIONAL,
        servingAreaID                   Lata                            OPTIONAL,
        serviceAddressInformationTriggerType    TriggerCriteriaType     OPTIONAL,
        callingPartyNumber              CallingPartyID                  OPTIONAL,
        callingPartysCategory           ChargePartyStationType          OPTIONAL,
        carrier                         Carrier                         OPTIONAL,
        cGEncountered                   ACGEncountered                  OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        sap                             Sap                             OPTIONAL,
        aMASequenceNumber               AMASequenceNumber               OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL,
        serviceAddressInformationMiscCallInfo   NotificationIndicator   OPTIONAL
        …
        }

OSuspendedArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey     UserID,
        bearerCapability                BearerCapability,
```

```
          serviceAddressInformationMiscCallInfo    NotificationIndicator      OPTIONAL,
          amp1                                     Amp1                       OPTIONAL,
          amp2                                     Amp2                       OPTIONAL,
          extensionParameter                       ExtensionParameter         OPTIONAL
          …
          }

OTermSeizedArg  ::= SEQUENCE {
          serviceAddressInformationServiceKey      UserID,
          bearerCapability                         BearerCapability,
          serviceAddressInformationMiscCallInfo    NotificationIndicator      OPTIONAL,
          amp1                                     Amp1                       OPTIONAL,
          amp2                                     Amp2                       OPTIONAL,
          extensionParameter                       ExtensionParameter         OPTIONAL
          …
          }

ReportErrorArg ::= SEQUENCE{
          applicationErrorString                   ApplicationErrorString,
          extensionParameter                       ExtensionParameter         OPTIONAL
          …
          }

ReleaseCallArg  ::= SEQUENCE {
          primaryBillingIndicator       PrimaryBillingIndicator                     OPTIONAL,
          aMAAlternateBillingNumber     AMAAlternateBillingNumber                   OPTIONAL,
          aMABusinessCustomerID         AMABusinessCustomerID                       OPTIONAL,
          aMALineNumber                 SEQUENCE SIZE (1..2) OF AMALineNumber       OPTIONAL,
          aMAslpID                      AMAslpID                                    OPTIONAL,
          aMADigitsDialedWC             SEQUENCE SIZE (1..2) OF AMADigitsDialedWC   OPTIONAL,
          amp1                          Amp1                                        OPTIONAL,
          amp2                          Amp2                                        OPTIONAL,
          serviceProviderID             ServiceProviderID                           OPTIONAL,
          serviceContext                ServiceContext                              OPTIONAL,
          aMABillingFeature             AMABillingFeature                           OPTIONAL,
          aMASequenceNumber             AMASequenceNumber                           OPTIONAL,
          extensionParameter            ExtensionParameter                          OPTIONAL,
          aMAserviceProviderID          AMAServiceProviderID                        OPTIONAL
          …
          }

RequestCurrentStatusReportArg ::= ResourceID
RequestNotificationChargingEventArg ::= SEQUENCE SIZE(1..numOfChargingEvents)
          OF ChargingEvent
RequestReportBCMEventArg ::= SEQUENCE {
          eDPRequest                    EDPRequest                                  OPTIONAL,
          eDPNotification               EDPNotification                             OPTIONAL,
          oNoAnswerTimer                ONoAnswerTimer                              OPTIONAL,
          tNoAnswerTimer                TNoAnswerTimer                              OPTIONAL,
          extensionParameter            ExtensionParameter                          OPTIONAL,
          timeoutTimer                  TimeoutTimer                                OPTIONAL,
          oDTMFDigitsString             ODTMFDigitsString                           OPTIONAL,
```

```
        oDTMFNumberOfDigits             ODTMFNumberOfDigits                          OPTIONAL,
        monitoringState                 MonitoringState                              OPTIONAL
        …
        }

ResetTimerArg ::= SEQUENCE {
        timerID                         [0] TimerID                                  DEFAULT tssf,
        timervalue                      [1] TimerValue,
        extensions                      [2] SEQUENCE SIZE(1..numOfExtensions)        OF
                                        ExtensionParameter                           OPTIONAL
        …
        }

ResourceClearArg ::= SEQUENCE {
        clearCause                      ClearCause,
        dialedDigitsCD                  CollectedDigits                              OPTIONAL,
        dialedDigitsCAI                 CollectedAddressInfo                         OPTIONAL,
        carrier                         Carrier                                      OPTIONAL,
        failureCause                    FailureCause                                 OPTIONAL,
        amp1                            Amp1                                         OPTIONAL,
        amp2                            Amp2                                         OPTIONAL,
        aMAMeasurement                  AMAMeasurement                               OPTIONAL,
        clearCauseData                  ClearCauseData                               OPTIONAL,
        iPReturnBlock                   IPReturnBlock                                OPTIONAL,
        extensionParameter              ExtensionParameter                           OPTIONAL
        …
        }

RouteSelectFailureArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey     UserID,
        bearerCapability                BearerCapability,
        chargeNumber                    ChargeNumber                                 OPTIONAL,
        servingAreaID                   Lata                                         OPTIONAL,
        serviceAddressInformationTriggerType    TriggerCriteriaType                  OPTIONAL,
        callingPartyNumber              CallingPartyID                               OPTIONAL,
        callingPartysCategory           ChargePartyStationType                       OPTIONAL,
        calledPartyNumber               CalledPartyID                                OPTIONAL,
        travellingClassMark             Tcm                                          OPTIONAL,
        originalCalledPartyID           OriginalCalledPartyID                        OPTIONAL,
        carrier                         Carrier                                      OPTIONAL,
        redirectingPartyID              RedirectingPartyID                           OPTIONAL,
        redirectionInformation          RedirectionInformation                       OPTIONAL,
        cGEncountered                   ACGEncountered                               OPTIONAL,
        amp1                            Amp1                                         OPTIONAL,
        amp2                            Amp2                                         OPTIONAL,
        sap                             Sap                                          OPTIONAL,
        aMASequenceNumber               AMASequenceNumber                            OPTIONAL,
        extensionParameter              ExtensionParameter                           OPTIONAL,
        serviceAddressInformationMiscCallInfo   NotificationIndicator                OPTIONAL
        …
        }
```

```
-- For the OPTIONAL parameters, See 7.4.6  for the trigger detection point processing
-- rules to specify when these parameters are included in the message.

SendChargingInformationArg ::= SEQUENCE {
        sCIBillingChargingCharacteristics   [0] SCIBillingChargingCharacteristics,
        partyToCharge                  [1] LegID,
        extensions                     [2] SEQUENCE SIZE(1..numOfExtensions)     OF
                                       ExtensionParameter                        OPTIONAL
        …
        }

SendNotificationArg ::= SEQUENCE{
        correlationID                  EchoData,
        extensionParameter             ExtensionParameter                        OPTIONAL
        …
        }

SendToResourceArg  ::= SEQUENCE {
        resourceType                   ResourceType,
        strParameterBlock              StrParameterBlock,
        disconnectFlag                 DisconnectFlag                            OPTIONAL,
        answerIndicator                AnswerIndicator                           OPTIONAL,
        primaryBillingIndicator        PrimaryBillingIndicator                   OPTIONAL,
        aMAAlternateBillingNumber      AMAAlternateBillingNumber                 OPTIONAL,
        aMABusinessCustomerID          AMABusinessCustomerID                     OPTIONAL,
        AMALineNumber                  SEQUENCE SIZE (1..2)                      OF
                                       AMALineNumber                             OPTIONAL,
        aMAslpID                       AMAslpID                                  OPTIONAL,
        aMADigitsDialedWC              SEQUENCE SIZE (1..2)                      OF
                                       AMADigitsDialedWC                         OPTIONAL,
        amp1                           Amp1                                      OPTIONAL,
        amp2                           Amp2                                      OPTIONAL,
        iPRoutingAddress               DestinationAddress                        OPTIONAL,
        dpConverter                    DPConverter                               OPTIONAL,
        aMAMeasure                     AMAMeasure                                OPTIONAL,
        serviceProviderID              ServiceProviderID                         OPTIONAL,
        serviceContext                 ServiceContext                            OPTIONAL,
        aMABillingFeature              AMABillingFeature                         OPTIONAL,
        aMASequenceNumber              AMASequenceNumber                         OPTIONAL,
        extensionParameter             ExtensionParameter                        OPTIONAL,
        aMAServiceProviderID           AMAServiceProviderID                      OPTIONAL
        …
        }

ServiceFilteringResponseArg ::= SEQUENCE {
        countersValue                  [0] CountersValue,
        filteringCriteria              [1] FilteringCriteria,
        extensions                     [2] SEQUENCE SIZE(1..numOfExtensions)     OF
                                       [ExtensionParameter                       OPTIONAL,
        responseCondition              [3] ResponseCondition                     OPTIONAL
        …
        }
```

```
SpecializedResourceReportArg ::=   NULL
StatusReportedArg ::= SEQUENCE{
        resourceStatus              FacilityStatus                      OPTIONAL,
        reportCondition             StatusCause                         OPTIONAL,
        failureCause                FailureCause                        OPTIONAL,
        amp1                        Amp1                                OPTIONAL,
        amp2                        Amp2                                OPTIONAL,
        extensionParameter          ExtensionParameter                  OPTIONAL
        …
        }

TAnswerArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey     UserID,
        bearerCapability            BearerCapability,
        serviceAddressInformationMiscCallInfo   NotificationIndicator   OPTIONAL,
        amp1                        Amp1                                OPTIONAL,
        amp2                        Amp2                                OPTIONAL,
        extensionParameter          ExtensionParameter                  OPTIONAL
        …
        }

TBusyArg  ::= SEQUENCE {
        serviceAddressInformationServiceKey     UserID,
        bearerCapability            BearerCapability,
        calledPartyNumber           CalledPartyID                       OPTIONAL,
        servingAreaID               Lata                                OPTIONAL,
        serviceAddressInformationTriggerType    TriggerCriteriaType     OPTIONAL,
        chargeNumber                ChargeNumber                        OPTIONAL,
        callingPartyNumber          CallingPartyID                      OPTIONAL,
        callingPartysCategory       ChargePartyStationType              OPTIONAL,
        originalCalledPartyID       OriginalCalledPartyID               OPTIONAL,
        redirectingPartyID          RedirectingPartyID                  OPTIONAL,
        redirectionInformation      RedirectionInformation              OPTIONAL,
        busyCause                   BusyCause                           OPTIONAL,
        busyType                    BusyType                            OPTIONAL,
        calledPartysCategory        CalledPartyStationType              OPTIONAL,
        sap                         Sap                                 OPTIONAL,
        displayInformationGN        GenericName                         OPTIONAL,
        serviceAddressInformationMiscCallInfo   NotificationIndicator   OPTIONAL,
        cGEncountered               ACGEncountered                      OPTIONAL,
        amp1                        Amp1                                OPTIONAL,
        amp2                        Amp2                                OPTIONAL,
        sTRConnection               STRConnection                       OPTIONAL,
        aMASequenceNumber           AMASequenceNumber                   OPTIONAL,
        extensionParameter          ExtensionParameter                  OPTIONAL
        …
        }

-- For the OPTIONAL parameters, See 7.4.6  for the trigger detection point processing rules
-- to specify when these parameters are included in the message.
```

```
TerminationAttemptArg ::= SEQUENCE {
        serviceAddressInformationServiceKey     UserID,
        bearerCapability                BearerCapability,
        calledPartyNumber               CalledPartyID                   OPTIONAL,
        servingAreaID                   Lata                            OPTIONAL,
        serviceAddressInformationTriggerType    TriggerCriteriaType     OPTIONAL,
        calledPartysCategory            CalledPartyStationType          OPTIONAL,
        chargeNumber                    ChargeNumber                    OPTIONAL,
        callingPartyNumber              CallingPartyID                  OPTIONAL,
        callingPartysCategory           ChargePartyStationType          OPTIONAL,
        travellingClassMark             Tcm                             OPTIONAL,
        originalCalledPartyID           OriginalCalledPartyID           OPTIONAL,
        redirectingPartyID              RedirectingPartyID              OPTIONAL,
        redirectionInformation          RedirectionInformation          OPTIONAL,
        displayInformationGN            GenericName                     OPTIONAL,
        cGEncountered                   ACGEncountered                  OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        sap                             Sap                             OPTIONAL,
        sTRConnection                   STRConnection                   OPTIONAL,
        aMASequenceNumber               AMASequenceNumber               OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL,
        …
        genericDigitsList               GenericDigitsList               OPTIONAL,
        callingGeodeticLocation         CallingGeodeticLocation         OPTIONAL
        …
        }

TerminationNotificationArg ::= SEQUENCE{
        correlationID                   EchoData,
        terminationIndicator            TerminationIndicator,
        connectTime                     ConnectTime                     OPTIONAL,
        busyCause                       BusyCause                       OPTIONAL,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL
        …
        }

TimeoutArg ::= SEQUENCE{
        serviceAddressInformationServiceKey     UserID,
        bearerCapability                BearerCapability,
        amp1                            Amp1                            OPTIONAL,
        amp2                            Amp2                            OPTIONAL,
        extensionParameter              ExtensionParameter              OPTIONAL,
        serviceAddressInformationMiscCallInfo   NotificationIndicator   OPTIONAL
        …
        }

TNoAnswerArg ::= SEQUENCE {
        serviceAddressInformationServiceKey     UserID,
        bearerCapability                BearerCapability,
```

```
        calledPartyNumber                   CalledPartyID                       OPTIONAL,
        servingAreaID                       Lata                                OPTIONAL,
        serviceAddressInformationTriggerType  TriggerCriteriaType               OPTIONAL,
        chargeNumber                        ChargeNumber                        OPTIONAL,
        callingPartyNumber                  CallingPartyID                      OPTIONAL,
        callingPartysCategory               ChargePartyStationType              OPTIONAL,
        originalCalledPartyID               OriginalCalledPartyID               OPTIONAL,
        redirectingPartyID                  RedirectingPartyID                  OPTIONAL,
        redirectionInformation              RedirectionInformation              OPTIONAL,
        calledPartysCategory                CalledPartyStationType              OPTIONAL,
        sap                                 Sap                                 OPTIONAL,
        displayInformationGN                GenericName                         OPTIONAL,
        serviceAddressInformationMiscCallInfo  NotificationIndicator            OPTIONAL,
        cGEncountered                       ACGEncountered                      OPTIONAL,
        amp1                                Amp1                                OPTIONAL,
        amp2                                Amp2                                OPTIONAL,
        sTRConnection                       STRConnection                       OPTIONAL,
        aMASequenceNumber                   AMASequenceNumber                   OPTIONAL,
        extensionParameter                  ExtensionParameter                  OPTIONAL
        …
        }

UpdateArg ::= SEQUENCE{
        administrableObject                 AdministrableObject,
        amp1                                Amp1                                OPTIONAL,
        amp2                                Amp2                                OPTIONAL,
        extensionParameter                  ExtensionParameter                  OPTIONAL
        …
        }

UpdateRequestArg ::= SEQUENCE{
        serviceAddressInformationServiceKey   UserID,
        triggerCriteriaFlag                 TriggerCriteriaFlag                 OPTIONAL,
        bearerCapability                    BearerCapability                    OPTIONAL
        …
        }
```

-- The Definition of Common Data Types Follows

AccessCode ::= [1] IMPLICIT INDigits

-- Range - 1-5 digits

ACGEncountered ::= [2] IMPLICIT OCTET STRING (SIZE (1))

```
-- ____________________________________________________________________
-- [        H        [       G      [  F  [  E  [  D  [  C  [  B  [  A  [
-- ____________________________________________________________________
-- [   SCP Overload  [  SMS Init   [          ACG Encountered           [
-- [    Cntrl Ind.   [  Cntrl Ind  [                                    [
-- ____________________________________________________________________
--
-- SCP Overload Controls Indicator
--
-- H
```

```
-- _____________________________________________________
-- 0              No SCP overload controls encountered
-- 1              SCP overload controls encountered
--
-- SMS Initiated Controls Indicator
--
-- G
-- ______________________________________________________
-- 0              No SMS initiated controls encountered
-- 1              SMS initiated controls encountered
--
-- ACG Encountered
--
-- FEDCBA
-- 000001         1-digit  control
-- . . .
-- 001010         10-digit control
```

AChBillingChargingCharacteristics ::= OCTET STRING (SIZE (minAChBillingChargingLength..
maxAChBillingChargingLength))

-- The AChBillingChargingCharacteristics parameter specifies the charging related information
-- to be provided by the SSF and the conditions on which this information has to be reported
-- back to the SCF with the ApplyChargingReport operation.
-- Examples of charging related information to be provided by the SSF may be: bulk counter
-- values, costs, tariff change and time of charge, time stamps, durations, etc.
-- Examples of conditions on which the charging related information are to be reported may be:
-- threshold value reached, timer expiration, tariff change, end of connection configuration, etc

ActivationStateCode ::= ENUMERATED {
-- A sub-field of the triggerItemAssignment field of the AdministrableObject parameter.

off (0),
on (1)
}

AdministrableObject ::= CHOICE {
triggerItemAssignment TriggerItemAssignment
sSPUserResource SSPUserResource
}

AdsiCpeID::=OCTET STRING (SIZE(4))

AlertingPattern ::= OCTET STRING (SIZE(3))
-- Indicates a specific pattern that is used to alert a subscriber (e.g. distinctive ringing, tones, etc.).
-- Only applies if SSF is the terminating local exchange for the subscriber. Refer to T1.607
-- Signal parameter for encoding.

AlternateBillingIndicator ::= [3] IMPLICIT BillingIndicator
-- This parameter is sent to indicate the AMA call type and service feature identifier
-- for the alternate trunk group.

AlternateCarrier ::= [4] IMPLICIT CarrierFormat
-- range - 4 digits

AlternateDialingPlanInd ::= [115] IMPLICIT INDigits
-- range 10 digits

AlternateTrunkGroup ::= [5] IMPLICIT OCTET STRING (SIZE (5))

```
-- _________________________________________________________________
-- [         H         [      G     [  F  [  E  [  D  [  C  [  B  [  A  [
-- _________________________________________________________________
-- [ No To  Outpulse [ SFG Ind  [           Call Treat Ind           [
-- _________________________________________________________________
-- [                Trunk Group Number (4 octets)                    [
-- _________________________________________________________________
--
-- Number to Outpulse
-- H
-- __________________________
-- 0   Outpulse number
-- 1   Normal routing number
--
-- Simulated Facility Group (SFG) Indicator
-- G
-- __________________________
-- 0   Not SFG
-- 1   SFG
--
-- Call Treatment Indicator (Reserved or not supported)
-- FEDCBA
-- _____________________________________________________
-- 000000    Reserved or not supported
--   to
-- 111111    Reserved or not supported
--
-- Route Index (8 BCD digits):
--    [ H [ G [ F [ E [ D [ C [ B [ A [
--    [_______________________________[
--    [    2nd Digit  [    1st Digit  [
--    [_______________________________[
--    [    4th Digit  [    3rd Digit  [
--    [_______________________________[
--    [    6th Digit  [    5th Digit  [
--    [_______________________________[
--    [    8th Digit  [    7th Digit  [
--    [_______________________________[
```

AMAAlternateBillingNumber ::= [6] IMPLICIT INDigits

-- *range 1-11 digits*
-- *This parameter is sent to identify an alternate billing number to which the IN service should be billed.*

AMABillingFeature

-- *This parameter is reserved for future use.*

AMABusinessCustomerID ::= [7] IMPLICIT INDigits

-- *range 1-11 digits*
-- *This parameter is sent to identify the business customer and the type of customer ID.*

AMADigitsDialedWC ::= [8] IMPLICIT INDigits

-- *range - 3-27 digits*
-- *In the context of the SEQUENCE OF AMADigitsDialedWC parameter, AMADigitsDialedWC is a field. This*
-- *field is sent to provide any digit string that the customer dialed along with a context ID to indicate the name of*
-- *the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).*

AMALineNumber ::= [9] IMPLICIT INDigits

-- *range - 3-15 digits*
-- *In the context of the SEQUENCE OF AMALineNumber parameter, AMALineNumber is a field. This field may*
-- *include information, such as the calling party ID, incoming terminating number, or Automatic Number*
-- *Identification (ANI).*

AMAMeasure ::= [71] IMPLICIT ENUMERATED {
connectTimeRecordedDestinationSSP (0),
connectTimeRecordedDestinationSCP (1),
connectTimeNotRecorded (2)
}

AMAMeasurement ::= [73] IMPLICIT SEQUENCE {
AMATimeDuration [0] IMPLICIT AMATimeDuration,
AMATimeGuard [1] IMPLICIT AMATimeGuard
}

AMASequenceNumber ::= [89] IMPLICIT OCTET STRING (SIZE(2))

-- *The AMASequenceNumber parameter shall be encoded using BCD as follows:*

--	*H*	*G*	*F*	*E*	*D*	*C*	*B*	*A*
--	*2nd Digit*				*1st Digit*			
--	*0000*				*3rd Digit*			

-- *This parameter is sent to indicate the order in which IN triggers are invoked within the context of an*
-- *originating or terminating call portion.*

AMAServiceProviderID ::= [101] IMPLICIT OCTET STRING (SIZE(7))

-- *This parameter contains a Context ID field, followed by a Service Provider Identifier field.*
-- *The Context ID field shall be encoded using BCD as follows:*

--	*H*	*G*	*F*	*E*	*D*	*C*	*B*	*A*
--	*2nd Digit*				*1st Digit*			
--	*0000*				*3rd Digit*			

-- *The Service Provider Identifier field shall be encoded using BCD as follows:*

--	*H*	*G*	*F*	*E*	*D*	*C*	*B*	*A*
--	*2nd Digit*				*1st Digit*			
--	*4th Digit*				*3rd Digit*			
--	*6th Digit*				*5th Digit*			
--	*8th Digit*				*7th Digit*			
--	*0000*				*9th Digit*			

-- *This parameter is sent to provide the identifier of a service provider when the service logic resides on an*
-- *SCP and the service provider is not the company whose equipment is generating the AMA.*

AMAslpID ::= [10] IMPLICIT OCTET STRING (SIZE(5))

--	*H*	*G*	*F*	*E*	*D*	*C*	*B*	*A*

--	*2nd Digit*	*1st Digit*
--	*4th Digit*	*3rd Digit*
--	*6th Digit*	*5th Digit*
--	*8th Digit*	*7th Digit*
--	*0000*	*9th Digit*

-- *This parameter is sent to indicate that the SSP should override normal switch-based recording and invoke*
-- *IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.*
-- *The most significant digit is sent first. Subsequent digits are sent in successive 4-bit fields.*

AMATimeDuration ::= OCTET STRING (SIZE(12))

```
-- A field of the AMAMeasurement parameter.
-- [ H [ G [ F [ E [ D [ C [ B [ A [
-- _________________________________
-- [     Month       [     Year       [
-- _________________________________
-- [     Day         [     Month      [
-- _________________________________
-- [ Filler (0000)  [     Day        [
-- _________________________________
-- [     Hour        [     Hour       [
-- _________________________________
-- [     Minute      [     Minute     [
-- _________________________________
-- [     Second      [     Second     [
-- _________________________________
-- [ Filler (0000)  [Tenth of Second[
-- _________________________________
-- [     Minute      [ Filler (0000) [
-- _________________________________
-- [     Minute      [     Minute     [
-- _________________________________
-- [     Minute      [     Minute     [
-- _________________________________
-- [     Second      [     Second     [
-- _________________________________
-- [ Filler (0000)  [Tenth of Second[
-- _________________________________
```

AMATimeGuard ::= ENUMERATED {
-- A field of the AMAMeasurement parameter.
noTimingGuard (0),
timingGuardExists (1)
}

Amp1 ::= [11] IMPLICIT OCTET STRING (SIZE(6))

-- *This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the*
-- *network.*

```
-- ______________________________________________________________________________
-- [    H    [    G    [    F    [    E    [    D    [    C    [    B    [    A    [
-- ______________________________________________________________________________
-- [ Reserved                                    [       CLogLev      [ Reserved[DoNotAlrt[
-- ______________________________________________________________________________
-- [                        AMP C Log Serial Number (most significant)            [
-- ______________________________________________________________________________
-- [                        AMP C Log Serial Number (least significant)           [
-- ______________________________________________________________________________
-- [                                                                              [
-- [______________                                              ________________[
```

```
-- [                                   AMP Time                             [
-- [_____________                                              _______________[
-- [                                                                          [
-- ____________________________________________________________________________
--
-- AMP Do Not Alert
-- A
-- ______________________________
-- 0      Alert call
-- 1      Do not alert call
--
-- Indicator (Reserved or Not Supported)
-- B
-- ______________________________________________________________
-- 0      Reserved or not supported
-- 1      Reserved or not supported
--
-- AMP C Log Level
-- DC
-- ________________________________________________________________________________
-- 00     C-Log not requested
-- 01     Generate time-stamped IN Network Test-related messages
-- 10     Generate time-stamped IN Network Test-related and IN system-internal
--        messages.
-- The AMP C Log Serial Number field shall be a 2 octet integer that identifies
-- a C-Log (along with the identifier of the IN Network Test originator). The
-- most significant octet is 00000000 when the AMP C Log Serial Number is less
-- than 256.
--
-- AMP Time
--   [   H   [   G   [   F   [   E   [   D   [   C   [   B   [   A   [
--   [_______________________________________________________________[
--   [ Null Ind.     [             Month             [     Year      [
--   [_______________________________________________________________[
--   [    spare              [           Date                        [
--   [_______________________________________________________________[
--   [ spare [    Min        [               Hour                    [
--   [_______________________________________________________________[
--
-- AMP Time Year
--  BA
-- _____________________
--  00      0 (last)
--  01      1 (current)
--  10      2 (next)
--
-- AMP Time Month
--  FEDC
--  0001   January
--  0010   February
--  0011   March
--  0100   April
--  0101   May
--  0110   June
--  0111   July
--  1000   August
--  1001   September
--  1010   October
--  1011   November
--  1100   December
--
-- AMP Time Null Indicator
--  HG
--  00     Null
--  01     Not null
--
-- AMP Time Date
--  EDCBA
--  00001  1
--  . . .
--  11111  31
--
```

```
-- AMP Time Hour
--
--  EDCBA
--  00000  0
--  . . .
--  10111  23
--
-- AMP Time Minute
--  GF
--  00     0 minutes
--  01     15 minutes
--  10     30 minutes
--  11     45 minutes
```

Amp2 ::= [109] IMPLICIT OCTET STRING

AnnounceElement ::= OCTET STRING (SIZE(3..131))

```
--   [  H  [  G  [  F  [  E [  D  [  C  [  B  [  A  [
--   _________________________________________________
--   [     AnnouncementID (most significant octet)  [
--   [     AnnouncementID (least significant octet) [
--   [              NumberOfInfoDigits              [
--   [     2nd Info Digit   [      1st Info Digit   [
--   [           *          [            *          [
--   [           *          [            *          [
--   [           *          [            *          [
--   [     Nth Info Digit   [     N-1st Info Digit  [
--   _________________________________________________
--
-- Info Digits
--   BCD    Meaning
--   0000   Digit 0 or filler
--   . . .
--   1001   Digit  9
--   1101   *
--   1110   #
--   1111   ST
```

AnnouncementBlock ::= SEQUENCE {

UninterAnnounceBlock	**[1] IMPLICIT UninterAnnounceBlock**	**OPTIONAL,**
InterAnnounceBlock	**[2] IMPLICIT InterAnnounceBlock**	**OPTIONAL**

}

AnnouncementDigitBlock ::= SEQUENCE {

MaximumDigits	**[0] IMPLICIT MaximumDigits,**	
UninterAnnounceBlock	**[1] IMPLICIT UninterAnnounceBlock**	**OPTIONAL,**
InterAnnounceBlock	**[2] IMPLICIT InterAnnounceBlock**	**OPTIONAL**

}

AnswerIndicator ::= [12] IMPLICIT NULL

-- Presence = return answer supervision
-- Absence = do not return answer supervision
-- This parameter is sent to indicate whether or not the SSP should return answer supervision to the
-- caller when the caller is connected to the announcement during the Send_To_Resource procedures.

ApplicationErrorString ::= [55] IMPLICIT SEQUENCE {

errorCause	**ErrorCause,**	
failedMessage	**FailedMessage**	**OPTIONAL,**
userID	**UserID**	**OPTIONAL**

}

-- This parameter is sent to identify the application error detected and may include information
-- pertaining to the erroneous message.

```
BearerCapability ::= [13] IMPLICIT ENUMERATED {
        speech                  (0),
        f31kHzaudio             (1),
        f7kHzaudio              (2),
        b56kbps                 (3),
        b64kbps                 (4),
        packetModeData          (5),
        multiRate               (6) -- (64 kbit/s base)
        }
```

-- packetModeData, 7KHz audio, and multiRate will NOT be used for T1 IN.

BillingIndicator ::= OCTET STRING (SIZE(4))

BusinessGroup ::= OCTET STRING (SIZE(7))

```
-- [  H   [  G   [   F   [  E    [  D   [   C   [   B  [   A  [
-- [_________________________________________________________[
-- [ Spare[ AttSt[BGID tp[LPII   [         Party Selector     [
-- [_________________________________________________________[
-- [msb*                                                      [
-- [_________                                      _________[
-- [            Business Group ID (3 octets)                  [
-- [_________                                      _________[
-- [                                                   lsb** [
-- [_________________________________________________________[
-- [msb*                                                      [
-- [_________                                      _________[
-- [                  Sub-Group ID (2 octets)          lsb** [
-- [_________________________________________________________[
-- [                  Line Privileges (1 octet)               [
-- [__________________________________________________________
--   *  msb = most significant bit
--   ** lsb = least significant bit
--
-- AttSt (Attendant Status)
--    G
--    0  No indication
--    1  Attendant line
--
-- BGID tp (Business Group Identifier Type)
--    F
--    0  BG Identifier
--    1  Reserved
--
-- LPII (Line Privileges Information Indicator)
--    E
--    0  Fixed Line Privileges
--    1  Reserved or not supported
--
-- Party Selector (the number to which the Business Group information applies)
--    DCBA
--    0000    Reserved or not supported
--    0001    Calling Party Number
--    0010    Reserved or not supported
--    0011    Reserved or not supported
--    0100    Redirecting Number
--    0101    Reserved or not supported
```

```
--   0110
--   to      Spare
--   1111
--
-- Business Group ID
--   0000..0000   Reserved or not supported
--   0000..0001   Public Network
--   0000..0010
--     to         Assigned Business Group Codes
--   1111..1111
--
-- Sub-Group ID
--   0000..0000    No Indication
--   0000..0001
--       to        Sub-Group IDs
--   1111..1111
--
-- Line Privileges  (of the party identified by the Party Selector)
-- If the LPII field is set to "Fixed Line Privileges", the LP field
-- is divided into two subfields to represent the terminating (bits ABCD)
-- and originating (bits EFGH) restrictions respectively.
--
--  Originating Restrictions:
--   HGFE
--   0000     Unrestricted
--   0001     Reserved or not supported
--   0010     Reserved or not supported
--   0011     Reserved or not supported
--   0100     Reserved or not supported
--   0101
--   to       Spare
--   1111
--
--  Terminating    Restrictions:
--   DCBA
--   0000     Unrestricted
--   0001     Reserved or not supported
--   0010     Reserved or not supported
--   0011     Reserved or not supported
--   0100     Reserved or not supported
--   0101
--   to       Spare
--   1111
--
-- If the LPII field is set to "1  Reserved or not supported"
-- the Line Privileges field is encoded as follows:
--
--   HGFEDCBA
--   00000000      Reserved or not supported
--   00000001
--      to         Reserved or not supported
--   11111111
```

BusyCause ::= [14] IMPLICIT OCTET STRING (SIZE(2..3))

-- *This parameter is sent to indicate the reason the call could not be presented to the terminating party. The*
-- *content of the BusyCause parameter is equivalent to the content of the ISUP Cause Indicators parameter.*
-- *The identifier of this parameter is different from the identifier of the ISUP Cause Indicators parameter. For all*
-- *fields in the BusyCause parameter, all missing field values should be considered spare in the tables that*
-- *follow.*

```
--
-- [   H   [   G   [   F   [   E   [   D  [   C   [   B   [   A   [
-- _______________________________________________________________
-- [   1   [ Coding        [ Spare [              General         [
-- [  ext  [ Standard      [       [              location        [
-- _______________________________________________________________
-- [  0/1  [                   Cause value                        [
```

```
-- [_______[                                                   [
-- [  ext  [      (class)          [      (value in class)         [
-- ___________________________________________________________________
-- [                      Diagnostics (if any)                      [
-- [                                                                [
-- ___________________________________________________________________
--
-- All missing field values should be considered spare.
--
-- Coding Standard
--   GF
--   00  CCITT-standard
--   11  Network-specific
--
-- General location
--   DCBA
--   0000   User
--   0001   Private network serving the local user
--   0010   Public network serving the local user
--   0011   Transit network
--   0100   Public network serving the remote user
--   0101   Private network serving the remote user
--   0110   Local interface controlled by this signaling link
--   0111   International network
--   1010   Unknown
--
-- Extension bit (in octet 2)
--   H
--   0  Diagnostic is included after octet 2 (see below)
--   1  Diagnostic is not included after octet 2
--
-- Cause value class
--   GFE
--   000   Normal event
--   001   Normal event
--   010   Resource unavailable
--   O11   Service or option not available
--   100   Service or option not implemented
--   101   Invalid message
--   110   Protocol error
--   111   Interworking
--
-- The Cause field shall be encoded as follows when the Coding Standard
-- field is set to "CCITT-standard" and the Cause value class is set to
-- "Normal event":
--   GFE    DCBA
--   000    0001 Unallocated number
--   000    0010 No route to specified transit network
--   000    0011 No route to destination
--   001    0000 Normal clearing
--   001    0001 User busy
--   001    0010 No user responding
--   001    0011 User alerting, no answer
--   001    0101 Call rejected
--   001    0110 Number changed
--   001    1001 Caller abandon
--   001    1010 Improper caller response
--   001    1011 Destination out of order
--   001    1100 Address incomplete
--   001    1111 Normal - unspecified (default, normal event class)
--   001    1101 Redirection counter exceeded
--
-- The Cause field shall be encoded as follows when the Coding Standard
-- field is set to "CCITT-standard" and the Cause value class is set to
-- "Resource unavailable":
--   GFE   DCBA
--   010   0010  No circuit available
--   010   0110  Reserved for network out of order
--   010   1001  Temporary failure
--   010   1010  Switching equipment congestion
--   010   1011  Access information discarded
--   010   1100  Reserved for requested channel not available
```

```
--   010   1111  Resource unavailable - unspecified (default, this class)
--
-- The Cause field shall be encoded as follows when the Coding Standard
-- field is set to "CCITT-standard" and the Cause value class is set to
-- "Service or option not available":
--   GFE   DCBA
--   O11   1001  Reserved for bearer capability not authorized
--   O11   1010  Bearer capability not presently available
--   O11   1111  Service/option not available - unspecified (default, this class)
--
-- The Cause field shall be encoded as follows when the Coding Standard
-- field is set to "CCITT-standard" and the Cause value class is set to
-- "Service or option not implemented":
--   GFE DBCA
--   100 0001      Bearer capability not implemented
--   100 1111      Service/option not implemented - unspecified (default, this class)
--
-- The Cause field shall be encoded as follows when the Coding Standard
-- field is set to "CCITT-standard" and the Cause value class is set to
-- "Invalid message":
--   GFE DCBA
--   101 1111     Invalid message - unspecified (default, this class)
--
-- The Cause field shall be encoded as follows when the Coding Standard
-- field is set to "CCITT-standard" and the Cause value class is set to
-- "Protocol error":
--   GFE    DCBA
--   110    0001  Message type non-existent or not implemented
--   110    0011  Reserved for parameter non-existent or not implemented - accepted
--   110    0100  Parameter with invalid coding
--   110    0101  Reserved for parameter non-existent or not implemented - discarded
--   110    1111  protocol error - unspecified (default, this class)
--
-- The Cause field shall be encoded as follows when the Coding Standard
-- field is set to "CCITT-standard" and the Cause value class is set to
-- "Interworking error:
--   GFE DCBA
--   111 1111    Interworking - unspecified (default, this class)
--
-- For a cause in class 110, Protocol error, a diagnostic may be included.
-- If included it should consist of the applicable ISUP parameter or
-- ISUP message name octet.
```

BusyType ::= [94] IMPLICIT ENUMERATED {

-- This parameter is sent to indicate whether or not the busy condition can be interrupted (e.g., with a
-- tone).

callCanBeOffered (0),
callCannotBeOffered (1)
}

CalledPartyBusinessGroupID ::= OCTET STRING

-- Indicates the business group of the callied party. The value is network operator-specific.

CalledPartyID ::= [15] IMPLICIT INDigits

-- Range - 0-15 digits

CalledPartyStationType ::= [16] IMPLICIT INTEGER (0..99)

-- Based on the Originating Line Information ISUP parameter.
-- Identified Line - No Special Treatment 0
-- ONI (Multiparty) 1
-- ANI Failure (unavailable) 2
-- Hotel (without room identification) 6
-- Coinless, Hospital, Inmate, etc. 7
-- InterLATA Restricted 8

-- AIOD - Listed DN sent 20
-- Identified line (coin or no coin) 23
-- 800 24
-- Coin call 27
-- IN 64
-- InterLATA restricted - hotel line 68
-- InterLATA restricted - coinless line, etc. 78
-- Test Call 95
-- Note: These are all the OLI values support by IN (a subset of the OLI values, that are the binary
-- equivalent of the II digits administered by North American Numbering Plan Administration,
-- http://www.nanpa.com). In addition, IN uses one of the reserved values, defining value (64) as "IN."

CalledPartySubaddress ::= OCTET STRING
-- Indicates the Called Party Subaddress. Refer to T1.607 for encoding.

CallingGeodeticLocation ::= [162] IMPLICIT OCTET STRING (SIZE(8..13))
-- This parameter contains the latitude and longitude information of an Emergency Caller.
-- See T1.628 for Emergency Calling Service description and usage.

CallingPartyBGID ::= [17] IMPLICIT BusinessGroup

CallingPartyBusinessGroupID ::= OCTET STRING
-- Indicates the business group of the calling party. The value of this octet string is network
-- operator-specific.

CallingPartyID ::= [18] IMPLICIT INDigits
-- range - 3-15 digits

CallingPartySubaddress ::= OCTET STRING
-- Indicates the Calling Party Subaddress. Refer to T1.607 for encoding.

CallResult ::= OCTET STRING (SIZE (minCallResultLength..
maxCallResultLength))
-- This parameter provides the SCF with the charging related information previously requested
-- using the ApplyCharging operation. This shall include the partyToCharge parameter as
-- received in the related ApplyCharging operation to correlate the result to the request
-- The remaining content is network operator-specific.
-- Examples of charging related information to be provided by the SSF may be: bulk counter values,
-- costs, tariff change and time of change, time stamps, durations, etc.
-- Examples of conditions on which the charging related information are to be reported may be:
-- threshold value reached, timer expiration, tariff change, end of connection configuration, etc.

Carrier ::= [41] IMPLICIT CarrierFormat
-- range - 4 digits

CarrierFormat ::= OCTET STRING (SIZE(4))
```
-- Used by the Carrier, AlternateCarrier and SecondAlternateCarrier parameters.
-- [ H [ G [  F [  E [ D [ C [ B [ A [
-- ___________________________________
-- [           Carrier Selection       [
-- ___________________________________
-- [                                   [
-- _________              ___________
-- [           Carrier                 [
-- _________              ___________
```

```
-- [                ID                 [
-- ___________________________________
--
-- Carrier Selection
--   HGFEDCBA
--   00000000     No indication
--   00000001     Selected carrier identification code presubscribed and not input by
--                calling party
--   00000010     Selected carrier identification code presubscribed and input by
--                calling party
--   00000011     Selected carrier identification code presubscribed, no indication of
--                whether input by calling party
--   00000100     Selected carrier identification code not presubscribed and input
--                by calling party
--   00000101
--      to        Spare
--   11111110
--   11111111     Reserved
--
-- Carrier ID
-- [ H [ G [ F [ E [ D [ C [ B [ A [
-- _________________________________
-- [       Number of Digits         [
-- [ 2nd Digit     [    1st Digit  [
-- [ 4th Digit     [    3rd Digit  [
-- _________________________________
```

CarrierUsage ::= [79] IMPLICIT ENUMERATED {

-- Indicates to the SSP how and when to use the carrier information contained
-- in the Carrier, AlternateCarrier and SecondAlternateCarrier parameters.

alwaysOverride (0),
onlyInterLATAOverride (1)
}

ChargeNumber ::= [19] IMPLICIT INDigits

-- Range - 0-10 digits

ChargePartyStationType ::= [20] IMPLICIT INTEGER (0..99)

-- Indicates the calling station type.
-- See CalledPartyStationType for explanation.

ClearCause ::= [21] IMPLICIT INTEGER (0..255)

```
-- Indicates why a connection between a user and a resource was terminated.
--  Value   Meaning
--  __________________________
--  0       normal
--  2       timeout
--  3       resourceCancelled
--  4       reserved or not supported
--  5       reserved or not supported
--  6       userAbandon
--  7       reserved or not supported
--  8       reserved or not supported
--  9       channelsBusy
--  10      calledPartyAnswered
--  11      resourceNotAvailable
--  12      isdnTimeout
--  13      resourceTypeNotSupported
--  14      taskRefused
--  15      invalidCallerResponse
--  16      capabilityFailure
--  17      protocolError
--  18      abort
--  19      suppServiceInvoked
```

```
--  20       strCancelled
--  21       temporaryFailure
--  22       ipTimeout
--  23       reserved or not supported
--  24       reserved or not supported
--  25       reserved or not supported
--  26       reserved or not supported
--  27       reserved or not supported
--  28       reserved or not supported
--  29       reserved or not supported
--  30       reserved or not supported
--  31       reserved or not supported
--  32-255   Reserved
```

Cause ::= OCTET STRING (SIZE (minCauseLength..maxCauseLength))

-- Indicates the cause for interface related information. Refer to T1.113
-- Cause parameter for encoding.

ChargingEvent ::= SEQUENCE {

```
        eventTypeCharging        [0] EventTypeCharging,
        monitorMode              [1] MonitorMode,
        legID                    [2] LegID                     OPTIONAL
        }
```

-- This parameter indicates the charging event type and corresponding
-- monitor mode and LegID

ClearCauseData ::= [74] IMPLICIT OCTET STRING (SIZE(1..20))

-- Provides information that may be returned from an IP in cases where
-- the IP terminates a connection with an ISDN Return Error component.

CloseCause ::= [72] IMPLICIT ENUMERATED {

-- Indicates the specific reason why a Close message is sent to
-- end a TCAP Transaction between the SSP and the SCP.

```
        callTerminated               (0),
        eDPsCompleted                (1),
        unexpectedCommunication      (2),
        calledPartyAnswered          (3)
        }
```

CollectedAddressInfo ::= [22] IMPLICIT INDigits

-- range 0 - 15 digits

CollectedDigits ::= [23] IMPLICIT INDigits

-- range 0-32 digits
-- Nature of Number field shall equal “Not applicable”
-- Numbering plan field shall equal “Unknown or not applicable”

ConnectTime ::= [58] IMPLICIT OCTET STRING (SIZE(5))

```
--    [ H  [  G  [  F  [  E  [ D  [  C  [  B  [  A  [
--   _________________________________________________
--           Minutes        [       Filler (0000)
--           Minutes        [       Minutes
--           Minutes        [       Minutes
--           Seconds        [       Seconds
--             0000         [       Tenths
```

ControlCauseIndicator ::= [59] IMPLICIT OCTET STRING (SIZE(1))

```
--     [          H          [         G          [ F [ E [ D [ C [ B [ A [
```

```
--
--      ____________________________________________________________________________
--      [ SCP Overload Cntrl Ind. [  SMS Init Cntrl Ind [      Number of Digits [
--      ____________________________________________________________________________
--
-- SCP Overload Controls Indicator
--    H
--    0     No SCP overload controls
--    1     SCP overload controls
--
-- SMS Initiated Controls Indicator
--    G
--    0    No SMS initiated controls
--    1    SMS initiated controls
--
-- Number of Digits to which Control is applied
--    FEDCBA
--    000001   reserved or not supported
--    000010   reserved or not supported
--    000011   3-digit control
--    000100   reserved or not supported
--    000101   reserved or not supported
--    000110   6-digit control
--    000111   7-digit control
--    001000   8-digit control
--    001001   9-digit control
--    001010   10-digit control
```

ControllingLegTreatment ::= [24] IMPLICIT ENUMERATED {

```
        dialToneOn                                  (0),
        ringBackAudibleRingingToneOn                (1),
        networkCongestionReorderToneOn              (3),
        busyToneOn                                  (4),
        confirmationTone                            (5),
        callWaitingTone                             (7),
        tonesOff                                    (63),
        alertingOnPattern0Normal                    (64),
        alertingOnPattern1DistinctiveIntergroup     (65),
        alertingOnPattern2DistinctiveSpecial        (66),
        alertingOnPattern3EKTS                      (67),
        alertingOnPattern4ReminderRing              (68),
        alertingOff                                 (79),
        recallDialToneOn                            (17),
        bargeInToneOn                               (18),
        incomingAdditionalCallTone                  (251),
        priorityAdditionalCallTone                  (252),
        expensiveRouteWarningTone                   (253),
        campOnTone                                  (19),
        receiverOffHookTone                         (20),
        callingCardServiceTone                      (21),
        stutterDialTone                             (22),
        silence                                     (23),
        onHookTR30WithIndication                    (24),
        onHookTR30NoIndication                      (25)
        }
```

CorrelationBlock ::=SEQUENCE{

```
        sCFCorrelationID              EchoData,
        sCFID                         SCFID
        }
```

CorrelationID ::= INDigits
-- used by SCF for correlation with a previous operation.See 7.4.6 for a description of the procedures
-- associated with this parameter.

```
CounterAndValue ::= SEQUENCE {
            counterID                   [0] CounterID,
            counterValue                [1] Integer4
            }
```

CounterID ::= INTEGER (0..99)
-- Indicates the counters to be incremented.
-- The counterIDs can be addressed by using the last digits of the dialed number.

CountersValue ::= SEQUENCE SIZE(0..numOfCounters) OF CounterAndValue

DateAndTime ::= OCTET STRING (SIZE(6))

```
-- Indicates, amongst others, the start time for activate service filtering. Coded as YYMMDDHHMMSS
-- with each digit coded BCD.
-- The first octet contains YY and the remaining items are sequenced following.
-- For example, 1993 September 30th, 12:15:01 would be encoded as:
-- Bits       HGFE     DCBA
-- leading octet3      9
--            9     0
--            0     3
--            2     1
--            5     1
--            1     0
```

DestinationAddress ::= [86] IMPLICIT INDigits
-- range - 10 digits

DestinationRoutingAddress ::= SEQUENCE SIZE(1..8) OF CalledPartyID
-- Indicates the list of Called Party Numbers (primary and alternates).

```
destinationRoutingAddressUsage ::= SEQUENCE {
            networkBusy                 [0],
            userBusy                    [1],
            noAnswer                    [2]BOOLEAN,
            noAnswerTimer               [3]OnoAnswerTimer
            }
-- Indicates which events should be monitored by the SSF.
-- and decide when to choose the next CalledPartyID from the DestinationRoutingAddress parameter.
```

DialingPlan ::= IA5String (SIZE(1..8))

```
DisconnectCause ::= [116] IMPLICIT ENUMERATED {
            farEnd                      (0)
            }
```

-- This parameter is sent to indicate the specific reason why an O_Disconnect message was sent to the SCP.

DisconnectFlag ::= [25] IMPLICIT NULL

-- This parameter is sent to indicate whether or not the SSP must disconnect a leg after a
-- Send_To_Resource
-- interaction is completed.
-- Presence = disconnect leg
-- Absence = do not disconnect leg

```
DisplayInformation ::= CHOICE {
        blank                       [0] IMPLICIT OCTET STRING (SIZE(1..20))
        skip                        [1] IMPLICIT OCTET STRING (SIZE(1..20))
        continuation                [2] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [3] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [4] IMPLICIT OCTET STRING (SIZE(1..20))
        progressIndicator           [5] IMPLICIT OCTET STRING (SIZE(1..20))
        notificationIndicator       [6] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [7] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [8] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [9] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [10] IMPLICIT OCTET STRING (SIZE(1..20))
        callingAddress              [11] IMPLICIT OCTET STRING (SIZE(1..20))
        reason                      [12] IMPLICIT OCTET STRING (SIZE(1..20))
        callingPartyName            [13] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [14] IMPLICIT OCTET STRING (SIZE(1..20))
        originalCalledName          [15] IMPLICIT OCTET STRING (SIZE(1..20))
        redirectingName             [16] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [17] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [18] IMPLICIT OCTET STRING (SIZE(1..20))
        dateTimeOfDay               [19] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [20] IMPLICIT OCTET STRING (SIZE(1..20))
        featureAddress              [21] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [22] IMPLICIT OCTET STRING (SIZE(1..20))
        redirectionNumber           [23] IMPLICIT OCTET STRING (SIZE(1..20))
        redirectingNumber           [24] IMPLICIT OCTET STRING (SIZE(1..20))
        originalCalledNumber        [25] IMPLICIT OCTET STRING (SIZE(1..20))
--   reserved or not supported   [26] IMPLICIT OCTET STRING (SIZE(1..20))
        text                        [30] IMPLICIT OCTET STRING (SIZE(1..20))
        redirectingReason           [31] IMPLICIT OCTET STRING (SIZE(1..20))
        }
--  Each DisplayInformation field can be from 1-20 octets.

DisplayText ::= [26] IMPLICIT SEQUENCE SIZE (1..15) OF DisplayInformation

Dn ::= OCTET STRING (SIZE(5))
-- The Dn will be 10 digits, encoded in BCD as follows:
--   _______________________________________________________________
--   [  H   [   G   [   F   [   E   [   D   [   C   [   B   [   A   [
--   _______________________________________________________________
--   [         2nd   digit         [          1st   digit          [
--   [         4th   digit         [          3rd   digit          [
--   [         6th   digit         [          5th   digit          [
--   [         8th   digit         [          7th   digit          [
--   [         10th  digit         [          9th   digit          [
--   _______________________________________________________________
-- The most significant digit is sent first.

DPConverter ::= [76] IMPLICIT BOOLEAN
--  This parameter is sent to indicate whether conversion from Dial Pulse to Dual Tone Multifrequency
--  (DTMF) is requested.
```

-- True = the switch should perform Dial Pulse to DTMF conversion
-- False = the switch should not perform Dial Pulse to DTMF conversion

DPNumber ::= INTEGER (1..33)

DTMFDigitsDetected ::= [153] IMPLICIT INDigits (SIZE(4))
-- This parameter is sent to indicate the actual DTMF digits collected.

Duration ::= INTEGER (-2..86400)
-- Values are in seconds. A duration of -1 indicates an infinite duration. A duration of -2 indicates
-- a network-specific duration.

EchoData ::= [60] IMPLICIT OCTET STRING (SIZE(6))

EDPNotification ::= [93] IMPLICIT BIT STRING {
-- This parameter indicates the requested EDP notifications.
-- Bit = 1 - notification is requested, Bit = 0 - notification is not requested

```
oCalledPartyBusy        (0), -- always "0"
oNoAnswer               (1), -- always "0"
oTermSeized             (2),
oAnswer                 (3),
tBusy                   (4), -- always "0"
tNoAnswer               (5), -- always "0"
termResourceAvailable   (6),
tAnswer                 (7),
networkBusy             (8), -- always "0"
oSuspended              (9), -- always "0"
oDisconnectCalled       (10), -- always "0"
oDisconnect             (11), -- always "0"
oAbandon                (12), -- always "0"
featureActivator        (13), -- always "0"
switchHookFlash         (14), -- always "0"
success                 (15), -- always "0"
tDisconnect             (16), -- always "0"
timeout                 (17), -- always "0"
originationAttempt      (18),
oDTMFEntered            (19)  -- always "0"
}
```

EDPRequest ::= [92] IMPLICIT BIT STRING {
-- This parameter indicates the requested EDP requests.
-- Bit = 1 - request is requested, Bit = 0 - request is not requested

```
oCalledPartyBusy        (0),
oNoAnswer               (1),
oTermSeized             (2), -- always "0"
oAnswer                 (3), -- always "0"
tBusy                   (4),
tNoAnswer               (5),
termResourceAvailable   (6), -- always "0"
tAnswer                 (7), -- always "0"
networkBusy             (8),
oSuspended              (9),
```

```
            oDisconnectCalled              (10),
            oDisconnect                    (11),
            oAbandon                       (12),
            featureActivator               (13),
            switchHookFlash                (14),
            success                        (15),
            tDisconnect                    (16),
            timeout                        (17),
            originationAttempt             (18),
            oDTMFEntered                   (19)
            }

EnvelopContent ::= [75] IMPLICIT OCTET STRING (SIZE(1..180))

EnvelopeEncodingAuthority ::= [98] IMPLICIT OBJECT IDENTIFIER -- Maximum of 15 octets

ErrorCause ::= [56] IMPLICIT ENUMERATED {
            erroneousDataValue             (0),
            missingConditionalParameter    (1),
            responseMessageTimerExpired    (2),
            unexpectedCommunication        (3),
            unexpectedMessage              (4),
            unexpectedMessageSequence      (5),
            unexpectedParameterSequence    (6)
            }
--  This parameter is sent to identify the application error detected at the SSF/SCF.

EventSpecificInformationCharging ::= OCTET STRING (SIZE (minEventSpecificInformationChargingLength..
                                 maxEventSpecificInformationChargingLength))
-- defined by network operator.
-- Indicates the charging related information specific to the event.

EventTypeCharging ::= OCTET STRING (SIZE (minEventTypeChargingLength..
                                 maxEventTypeChargingLength))
-- This parameter indicates the charging event type. Its content is network operator-specific.

ExtensionParameter ::= [84] IMPLICIT SEQUENCE {
            assignmentAuthority            OBJECT IDENTIFIER,
            parameters                     ANY DEFINED BY assignmentAuthority
            }

--    The maximum length of the assignmentAuthority is 15 octets.
--    The parameters field is to be defined by individual network operators.
--    This parameter is sent to provide network operator-specific information. It enables a network operator to add
--    parameters to a message sent over the SSP-SCP interface. This parameter contains two fields: Assignment
--    Authority and network operator-specific parameters. Unless specific arrangements are made with a switch
--    supplier, the contents of the network operator-specific parameters field cannot be processed by IN SSPs. If
--    the need arises for network operators to define parameters during the post-IN time frame, the parameters are
--    to be defined in the network operator-specific parameters field. When this parameter is included in a
--    message sent between the SCP and the SSP, the Assignment Authority field will be coded with the Object
--    Identifier corresponding to the network operator document that defines the contents of the network operator-
--    specific parameters field.
```

```
FacilityGID ::= CHOICE {
        trunkGroupID            TrunkGroupID
        privateFacilityGID      PrivateFacilityGID
        mlhg                    Mlhg
        routeIndex              RouteIndex
        }

FacilityMemberID ::= [31] IMPLICIT INTEGER (1..2047)

FacilityStatus ::= [61] IMPLICIT ENUMERATED {
        busy                    (0),
--  reserved or not supported     (1),
--  reserved or not supported     (2),
        idle                    (3),
--  reserved or not supported     (4),
--  reserved or not supported     (5),
--  reserved or not supported     (6),
--  reserved or not supported     (7)
        }

FailedMessage ::= [57] IMPLICIT SEQUENCE {
        opCode                  INTEGER,
        parameter               OCTET STRING                OPTIONAL,
        invParms                [1] IMPLICIT InvParms       OPTIONAL
        }

-- The opCode field is equivalent to the value of the Operation Code that represents the
-- failed message. The parameter field represents the parameter, including the identifier,
-- length, and contents, of the received message that contains the erroneous data value.
-- This parameter is sent to provide the Operation Code that was received when an
-- application error was detected.
-- This parameter may include the identifier, length, and contents of the parameter
-- associated with the received  message that contains the erroneous data value.

FailureCause ::= [32] IMPLICIT ENUMERATED {
        RateTooHigh             (1),
        UnavailableResources    (2),
--  reserved or not supported     (3),
--  reserved or not supported     (4),
--  reserved or not supported     (5),
--  reserved or not supported     (6),
--  reserved or not supported     (7),
--  reserved or not supported     (8),
--  reserved or not supported     (9),
--  reserved or not supported     (10),
--  reserved or not supported     (11),
--  reserved or not supported     (12),
--  reserved or not supported     (13),
        abort                   (14),
        resourceLimitation      (15),
        applicationError        (16),
--  reserved or not supported     (17),
            protocolError       (18),
            timerExpired        (19),
            temporaryFailure    (20),
--  reserved or not supported     (21),
```

```
                segmentationError       (22),
                ncasDisallowed          (23),
--   reserved or not supported          (24),
--   reserved or not supported          (25),
--   reserved or not supported          (26),
--   reserved or not supported          (27),
--   reserved or not supported          (28),
--   reserved or not supported          (29)
            }
--  This parameter is sent to indicate that the received operation could not be performed due to the
--  unavailability of a hardware or software resource (i.e., a failure occurred).

FailureCauseData ::= [112] IMPLICIT OCTET STRING (SIZE(1..5))
--  This parameter is sent to indicate the Error parameter found in a message with a Return Error
--  component that the SSF receives from the IP. This parameter is sent only if the Error parameter is

FCIBillingChargingCharacteristics ::= OCTET STRING (SIZE (minFCIBillingChargingLength..
                                maxFCIBillingChargingLength))
-- This parameter indicates the billing and/or charging characteristics. Its content is network operator-specific.

FeatureRequestIndicator ::= ENUMERATED {
            hold(0),
            retrieve(1),
            featureActivation(2),
            spare1(3),
            sparen(127)
            }
-- Indicates the feature activated (e.g. a switch-hook flash, feature activation). Spare values reserved
-- for future use.

FilteredCallTreatment ::= SEQUENCE {
            sFBillingChargingCharacteristics [0] SFBillingChargingCharacteristics,
            informationToSend               [1] InformationToSend               OPTIONAL,
            maximumNumberOfCounters         [2] MaximumNumberOfCounters         OPTIONAL,
            releaseCause                    [3] Cause                           OPTIONAL
            }
-- If releaseCause is not present, the default value is the same as the ISUP cause value decimal 31.
-- If informationToSend is present, the call will be released after the end of the announcement
-- with the indicated or default releaseCause.
-- If maximumNumberOfCounters is not present, ServiceFilteringResponse will be sent with
-- CountersValue::= SEQUENCE SIZE (0) OF CountersAndValue

FilteringCharacteristics ::= CHOICE {
            interval                        [0] INTEGER (-1..32000)
            numberOfCalls                   [1] Integer4
            }
-- Indicates the severity of the filtering and the point in time when the ServiceFilteringResponse is to be sent.
-- If = interval, every interval of time a ServiceFilteringResponse is sent to
-- the SCF. The interval is specified in seconds.
-- If = NumberOfCalls, every N calls for the Nth call a ServiceFilteringResponse
-- is sent to the SCF.
-- If ActivateServiceFiltering implies several counters - filtering on several dialled numbers -,
-- the numberOfCalls would include calls to all the dialled numbers.

FilteringCriteria ::= CHOICE {
```

```
        dialledNumber                [0] INDigits
        callingLineID                [1] INDigits
        serviceKey                   [2] UserID
        addressAndService            [30] SEQUENCE {
        calledAddressValue           [0] INDigits,
        serviceKey                   [1] UserID,
        callingAddressValue          [2] INDigits                 OPTIONAL,
        locationNumber               [3] LocationNumber           OPTIONAL
        }
        }
-- In case calledAddressValue is specified, the numbers to be filtered are from calledAddressValue
-- up to and including calledAddressValue + maximumNumberOfCounters-1.
-- The last two digits of calledAddressvalue can not exceed 100-maximumNumberOfCounters.

FilteringTimeOut ::= CHOICE {
        duration                     [0] Duration
        stopTime                     [1] DateAndTime
        }
-- Indicates the maximum duration of the filtering. When the timer expires, a ServiceFilteringResponse
-- is sent to the SCF.

FlexParameterBlock ::= OCTET STRING (SIZE(1..120))
-- The FlexParameterBlock allows resources to be defined in a flexible manner for all entities
-- (e.g., network providers), and may be used to access any IP resource, whether it is a new
-- resource or a basic one such as Play Announcement.
-- FlexParameterBlock contains such information as who encoded this parameter, what is the
-- actual resource to be accessed during this connection to the IP, and any additional information
-- for the capability specified.

ForwardCallIndicator ::= [113] IMPLICIT OCTET STRING (SIZE (2))
--  Defined per T1.113.3

GapDuration ::= [62] IMPLICIT ENUMERATED {
        no1Second                    (1),
        no2Seconds                   (2),
        no4Seconds                   (3),
        no8Seconds                   (4),
        no16Seconds                  (5),
        no32Seconds                  (6),
        no64Seconds                  (7),
        no128Seconds                 (8),
        no256Seconds                 (9),
        no512Seconds                 (10),
        no1024Seconds                (11),
        no2048Seconds                (12),
        infinity                     (13)
        }

GapInterval ::= CHOICE {
        NationalGapInterval          NationalGapInterval
        privateGapInterval           PrivateGapInterval
        }

GenericAddress ::= [80] IMPLICIT OCTET STRING (SIZE(4..11))
```

-- Defined per T1.113.3

GenericAddressList ::= [107] IMPLICIT SEQUENCE SIZE (1..5) OF GenericAddress

GenericDigits ::= [149] IMPLICIT OCTET STRING (SIZE(2..11))

```
-- [  H  [  G  [  F  [  E  [  D  [  C   [  B  [  A  [
-- __________________________________________________
-- [ Encoding Scheme [     Type of Digits           [
-- __________________________________________________
-- [                   Digits                       [
-- __________________________________________________
-- [                     *                          [
-- [                     *                          [
-- [                     *                          [
-- __________________________________________________
-- [                   Digits                       [
-- __________________________________________________
--
--
-- Encoding Scheme:
-   HGF
--  ______________
--  000  BCD Even
--  001  BCD Odd
--  010  IA5
--  011  Binary
--  100
--  to   Spare
--  111
--
-- For encoding of digits see Telcordia™ GR-1298-CORE and GR-1299-CORE.
```

GenericDigitsList ::= [150] IMPLICIT SEQUENCE SIZE (1..5) OF GenericDigits

GenericName ::= [33] IMPLICIT OCTET STRING

```
-- GenericName:
-- [  H  [  G  [  F  [   E   [  D  [  C  [   B   [   A   [
-- _______________________________________________________
-- [ Type of Name     [ Avail [ Spare     [ Presentation  [
-- _______________________________________________________
--
-- Presentation
--   BA
--   00   Presentation Allowed
--   01   Presentation Restricted
--   10   Blocking Toggle
--   11   No indication
--
-- Availability
--   E
--   0  Name available, or name availability not known
--   1  Name unavailable
--
-- Type of Name
--   HGF
--   001   Calling Name
--   010
--   to    spare
--   111
```

GlobalTitleAddress ::= [69] IMPLICIT OCTET STRING

```
-- GlobalTitleAddress:
--   [  H  [  G  [  F [  E  [  D  [  C  [  B  [  A  [
--  ________________________________________________
--       2nd Digit           [         1st Digit
--       4th Digit           [         3rd Digit
--       6th Digit           [         5th Digit
--       8th Digit           [         7th Digit
--       10th Digit          [         9th Digit
--  ________________________________________________
-- The most significant digit is sent first.
```

InbandInfo ::= SEQUENCE {

messageID	**[0] MessageID,**	
numberOfRepetitions	**[1] INTEGER (1..127)**	**OPTIONAL,**
duration	**[2] INTEGER (0..32767)**	**OPTIONAL,**
interval	**[3] INTEGER (0.. 32767)**	**OPTIONAL**

}

-- Interval is the time in seconds between each repeated announcement. Duration is the total
-- amount of time in seconds, including repetitions and intervals.
-- The end of announcement is either the end of duration or numberOfRepetitions, whatever comes first.
-- duration with value 0 indicates infinite duration

INDigits ::= OCTET STRING

```
-- Parameters that use the INDigits format are:
--   Calling Parameters: AlternateDialingPlanInd,
--   CallingPartyID, OriginalCalledPartyID, and
--   RedirectingPartyID.
--
--  Called Parameters: AccessCode, CalledPartyID,
--    CollectedAddressInfo, CollectedDigits, DestinationAddress, Lata,
--    OutpulseNumber, Tcm, and VerticalServiceCode.
--
--  Charge Number
--
-- __________________________________________________
-- [  H  [  G  [  F  [  E  [  D  [  C   [  B  [  A  [
-- __________________________________________________
-- [ Odd/[            Nature of Number              [
-- [ Even[                                          [
-- __________________________________________________
-- [Spare[ Numbering Plan  [Pres./Spare*[Scr./Spare*[
-- __________________________________________________
-- [    2nd Digit          [       1st Digit        [
-- __________________________________________________
-- [        *              [          *             [
-- [        *              [          *             [
-- [        *              [          *             [
-- __________________________________________________
-- [    Nth Digit          [       N-1st Digit      [
-- __________________________________________________
--
-- * These fields are spare for the Called and Charge Number
--   parameters.
--
-- The most significant digit is sent first. Subsequent digits are sent in
-- successive 4-bit fields.
--
-- Odd/Even Indicator (for Calling, Called and Charge Number parameters)
-- H
-- _____________________________________
-- 0             Even number of digits
-- 1             Odd number of digits
--
-- Nature of Number (for Called parameters)
-   GFEDCBA
--  __________________________________________________________________
```

```
--  0000000  Not applicable
--  0000001  Subscriber number
--  0000010  Spare, reserved for national use
--  0000011  National (significant) number
--  0000100  International number
--  0000101
--     to    Spare
--  1110000
--  1110001  Subscriber number, operator requested (0+ call)
--  1110010  National number, operator requested (0+ call)
--  1110011  International number, operator requested (01+ call)
--  1110100  No address present, operator requested (0-, 101XXXX+0(0), or 00- call)
--  1110101  No address present, cut-through call to carrier
--   1110110  950+ call from local exchange carrier public station or hotel/motel line
or
--           non-EAEO
--  1110111  Test Line Test code
--  1111000
--     to    Reserved for network-specific use
--  1111110

-- Nature of Number (for Calling parameters)
-- GFEDCBA
-- ______________________________________________________________________________
-- 0000000   Unknown or not applicable, default
-- 0000001   Unique subscriber number
-- 0000010   Spare, reserved for national use
-- 0000011   Unique national (significant) number
-- 0000100   Unique international number
-- 0000101
--   to      Spare
-- 1110000
-- 1110001   Non-unique subscriber number
-- 1110010   Spare, reserved for national use
-- 1110011   Non-unique national number
-- 1110100   Non-unique international number
-- 1110101   Spare
-- 1110011   Spare
-- 1110111   Test Line Test code
-- 1111000
--   to      Reserved for network-specific use
-- 1111110
-- 1111111   Reserved
--
-- Nature of Number (for Charge Number Parameters)
-- GFEDCBA
-- ____________________________________________________________________________
-- 0000000   Spare
-- 0000001   ANI of the calling party; subscriber number
-- 0000010   ANI not available or not provided
-- 0000011   ANI of the calling party; national number
-- 0000100   Spare
-- 0000101   ANI of the called party included; subscriber number
-- 0000110   ANI of the called party; not included
-- 0000111   ANI of the called party included; national number
-- 0001000
--    to     Spare
-- 1110111
-- 1111000
--    to     Reserved for network-specific use
-- 1111110
-- 1111111   Reserved
--
-- Numbering Plan (for Calling, Called and Charge Number parameters)
-- GFE
-- ______________________________________________________________________________
-- 000       Unknown or not applicable
-- 001       ISDN Numbering Plan (Rec. E.164)
-- 010
-- to        Reserved
-- 100
-- 101       Private
```

```
-- 110        Reserved
-- 111        Reserved
--
-- Presentation Restriction Indicator (for Calling Number parameters)
-- DC
-- ____________________________________________________________________________
-- 00         Presentation allowed
-- 01         Presentation restricted (default)
-- 10         Number unavailable
-- 11         Reserved
--
-- Screening Indicator (for Calling Number parameters)
-- BA
-- _______________________________________________________________________________
-- 00         Reserved for user provided, not screened or spare
-- 01         User provided, passed network screening
-- 10         Reserved for user provided, failed network screening
-- 11         Network provided
--
-- Digits
-- Binary Coded
-- Decimal (BCD)  Meaning
-- ____________________________________________________________________________
-- 0000       Digit 0 or Filler
-- . . .
-- 1001       Digit 9
-- 1010       Spare
-- 1011       Reserved
-- 1100       Reserved
-- 1101       *
-- 1110       #
-- 1111       ST
```

InformationToSend ::= CHOICE {
inbandInfo [0] InbandInfo
tone [1] Tone
displayInformation [2] DisplayInformation
}

Integer4 ::= INTEGER(0..2147483647)

InterAnnounceBlock ::= SEQUENCE (SIZE 1..10) OF AnnounceElement

IntervalTime ::= IMPLICIT INTEGER (1..86400)
-- Indicates the time interval (in seconds) for Timeout Timer.

InvParms ::= SEQUENCE SIZE (1) OF Parms

IPReturnBlock ::= [78] IMPLICIT OCTET STRING (SIZE(1..120))

ISDNAccessRelatedInformation ::= OCTET STRING
-- Indicates the destination user network interface related information. Refer to T1.113
-- Access Transport parameter for encoding.

Lata ::= [35] IMPLICIT INDigits
-- Range - 3 digits

LegID ::= CHOICE {
sendingSideID [0] LegType
receivingSideID [1] LegType

```
        }
-- Indicates a reference to a specific party in a call. OPTIONAL denotes network operator-specific use
-- with a choice of unilateral ID assignment or bilateral ID assignment.
-- OPTIONAL for LegID also denotes the following:
--   when only one party exists in the call, this parameter is not needed (as no ambiguity exists);
--   when more than one party exists in the call, one of the following alternatives applies:
--          1. LegID is present and indicates which party is concerned.
--          2. LegID is not present and a default value is assumed (e.g. calling party in the case of the
--             ApplyCharging operation).
-- Choice between these two alternatives is kept a network operator option.

LegType ::= OCTET STRING (SIZE(1))
leg1 LegType ::= '01'H
leg2 LegType ::= '02'H

LocalSSPTime ::= IMPLICIT OCTET STRING (SIZE(7))
-- LocalSSPTime:
-- [   H  [  G  [  F  [   E  [  D  [  C  [  B  [  A  [
-- [_________________________________________________[
-- [       Hours                [       Hours        [
-- [_________________________________________________[
-- [       Minutes              [       Minutes      [
-- [_________________________________________________[
-- [       Seconds              [       Seconds      [
-- [_________________________________________________[
-- [       Day                  [       Day          [
-- [_________________________________________________[
-- [       Month                [       Month        [
-- [_________________________________________________[
-- [       Year                 [       Year         [
-- [_________________________________________________[
-- [       Year                 [       Year         [
-- [_________________________________________________[

-- Indicates the time based on the local time zone of the SSP.

LocationNumber ::= OCTET STRING (SIZE (minLocationNumberLength..
                                        maxLocationNumberLength))
-- Indicates the Location Number for the calling party. Refer to ITU-T Recommendation Q.763 (White book)
-- for encoding.

MaximumDigits ::= INTEGER (0..255)
-- MaximumDigits:
-- range: 0-32 is “fixed number of digits” (e.g. 5 means collect 5 digits)
-- 33-252 is spare
-- 253 is “normal number of digits”
-- 254 is “variable number of digits”
-- 255 is spare

MaximumNumberOfCounters ::= INTEGER (1..numOfCounters)
MessageID ::= CHOICE {
        elementaryMessageID          [0] Integer4
        text                         [1] SEQUENCE {
        messageContent               [0] IA5String (SIZE (minMessageContentLength
        maxMessageContentLength)),
        attributes                   [1] OCTET STRING (SIZE (minAttributesLength
        maxAttributesLength))
        OPTIONAL
        }
```

elementaryMessageIDs [29] SEQUENCE SIZE
(1..numOfMessageIDs) OF Integer4
variableMessage [30] SEQUENCE {
elementaryMessageID [0] Integer4,
variableParts [1] SEQUENCE SIZE (1..5) OF VariablePart
}
}

-- OPTIONAL denotes network operator-specific use.

Mlhg ::= [29] IMPLICIT INTEGER (1..2047)

MonitoringState ::= SEQUENCE {
alerting [0],
active [1]
}

-- Indicates when the SSF should monitor the specified oDTMFEntered event

MonitorMode ::= ENUMERATED {
interrupted(0),
notifyAndContinue(1),
transparent(2)
}

-- Indicates the event is relayed and/or processed by the SSP.
-- If this parameter is used in the context of charging events, the following definitions apply for the
-- handling of charging events:
-- Interrupted means that the SSF notifies the SCF of the charging event using
-- EventNotificationCharging, does not process the event but discard it.
-- NotifyAndContinue means that SSF notifies the SCF of the charging event using
-- EventNotificationCharging, and continues processing the event or signal without waiting for SCF.
-- instructions Transparent means that the SSF does not notify the SCF of the event. This value is used to
-- end the monitoring of a previously requested charging event. Previously requested charging events are
-- monitored until ended by a transparent monitor mode, or until the end of the connection configuration.

MonitorTime ::= [65] IMPLICIT OCTET STRING (SIZE(3))

```
-- MonitorTime:
-- [  H [ G [ F [  E [ D [ C [ B [ A [
-- [__________________________________[
-- [     Hours        [      Hours    [
-- [__________________________________[
-- [    Minutes       [     Minutes   [
-- [__________________________________[
-- [    Seconds       [     Seconds   [
-- [__________________________________[
```

Mwi ::= [100] IMPLICIT ENUMERATED {
activated (0),
deactivated (1)
}

NationalGapInterval ::= [63] IMPLICIT ENUMERATED {
removeGapControl (0),
no0Seconds (1),
no010Seconds (2), *-- 1/10 second*
no025Seconds (3), *-- 1/4 second*
no050Seconds (4), *-- 1/2 second*

```
        no1Second                  (5),
        no2Seconds                 (6),
        no5Seconds                 (7),
        no10Seconds                (8),
        no15Seconds                (9),
        no30Seconds                (10),
        no60Seconds                (11),
        no120Seconds               (12),
        no300Seconds               (13),
        no600Seconds               (14),
        stopAllCalls               (15)
        }
```

NetworkSpecificFacilities ::= [108] IMPLICIT OCTET STRING (SIZE (2..12))
-- This parameter is sent to identify the particular service being delivered to the PRI.

NotificationIndicator ::= [111] IMPLICIT BOOLEAN
-- This parameter indicates whether the event is an EDR-N or an EDP-R.
-- TRUE = Switch Notification Message
-- FALSE = Switch Request Message

ODTMFDigitsString ::= [154] IMPLICIT INDigits
-- Range 1-4 digits.
-- This parameter is sent to indicate the digit sequence to be detected.

ODTMFNumberOfDigits ::= [155] IMPLICIT INTEGER(1..4)
-- This parameter is sent to indicate the number of DTMF digits to be detected.

ONoAnswerTimer ::= [91] IMPLICIT INTEGER (1..120)
-- This parameter is sent to indicate the value, in seconds, of the SSP
-- originating no answer timer.

OriginalCalledPartyID ::= [36] IMPLICIT INDigits
-- Range - 3-15 digits

OutpulseNumber ::= [37] IMPLICIT INDigits
-- 15 digits maximum

OverflowBillingIndicator ::= [38] IMPLICIT BillingIndicator
-- This parameter is sent to provide the AMA call type and service feature identifier for the carrier that is used to
-- route the call.

Parms `-- This parameter is reserved for future use.`

PassiveLegTreatment ::= [39] IMPLICIT ENUMERATED {

```
        dialToneOn                          (0),
        ringBackAudibleRingingToneOn        (1),
        networkCongestionReorderToneOn      (3),
        busyToneOn                          (4),
        confirmationTone                    (5),
        callWaitingTone                     (7),
```

```
            tonesOff                                  (63),
            alertingOnPattern0Normal                  (64),
            alertingOnPattern1DistinctiveIntergroup   (65),
            alertingOnPattern2DistinctiveSpecial      (66),
            alertingOnPattern3EKTS                    (67),
            alertingOnPattern4ReminderRing            (68),
            alertingOff                               (79),
            recallDialToneOn                          (17),
            bargeInToneOn                             (18),
            incomingAdditionalCallTone                (251),
            priorityAdditionalCallTone                (252),
            expensiveRouteWarningTone                 (253),
            campOnTone                                (19),
            receiverOffHookTone                       (20),
            callingCardServiceTone                    (21),
            stutterDialTone                           (22),
            silence                                   (23)
            }

PrimaryBillingIndicator ::= [40] IMPLICIT BillingIndicator
 --    This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group
 --    and for services when the primary trunk group is not provided.

PrimaryTrunkGroup ::= [42] IMPLICIT OCTET STRING (SIZE(5))
-- Uses same format as defined for AlternateTrunkGroup

PrivateFacilityGID ::= INTEGER (0..9999)

PrivateGapInterval ::=[64] IMPLICIT ENUMERATED {
            reserved or not supported        (0),
            no3Seconds                       (1),
            no4Seconds                       (2),
            no6Seconds                       (3),
            no8Seconds                       (4),
            no11Seconds                      (5),
            no16Seconds                      (6),
            no22Seconds                      (7),
            no30Seconds                      (8),
            no42Seconds                      (9),
            no58Seconds                      (10),
            no81Seconds                      (11),
            no112Seconds                     (12),
            no156Seconds                     (13),
            no217Seconds                     (14),
            no300Seconds                     (15),
            removeGapControl                 (16)
--  reserved or not supported         (17),
--  reserved or not supported         (18),
--  reserved or not supported         (19),
--  reserved or not supported         (20),
--  reserved or not supported         (21)
            }
```

RedirectingPartyID ::= [43] IMPLICIT INDigits
-- Range - 3-15 digits

RedirectionInformation ::= [44] IMPLICIT OCTET STRING (SIZE(2))
```
-- RedirectionInformation:
--   _____________________________________________________________
--   [   H   [   G   [   F   [   E   [   D  [  C  [   B   [   A   [
--   _____________________________________________________________
--   [  Original Redirecting Reason  [           Reserved         [
--   _____________________________________________________________
--   [      Redirecting Reason       [      Redirection Counter  [
--   _____________________________________________________________
--
-- Original Redirecting Reason and Redirecting Reason
--   HGFE
--   0000   unknown/not available
--   0001   user busy
--   0010   no reply
--   0011   unconditional
--   0100
--   to     spare
--   1111
--
-- Redirection Counter field shall be an integer (range 1-10).
```

RequestedInformationList ::= SEQUENCE SIZE (1..numOfInfoItems) OF RequestedInformation
RequestedInformationTypeList ::= SEQUENCE SIZE (1..numOfInfoItems) OF RequestedInformationType
RequestedInformation ::= SEQUENCE {
requestedInformationType [0] RequestedInformationType,
requestedInformationValue [1] RequestedInformationValue
}

RequestedInformationType ::= ENUMERATED {
callAttemptElapsedTime(0),
callStopTime(1),
callConnectedElapsedTime(2),
calledAddress(3),
releaseCause(30)
}

RequestedInformationValue ::= CHOICE {
callAttemptElapsedTimeValue [0] INTEGER (0..255)
callStopTimeValue [1] DateAndTime
callConnectedElapsedTimeValue [2] Integer4
calledAddressValue [3] INDigits
releaseCauseValue [30] Cause
}
-- The callAttemptElapsedTimeValue is specified in seconds. The unit for the
-- callConnectedElapsedTimeValue is 100 milliseconds

ResourceID ::= CHOICE {
lineID [0] INDigits
facilityGroupID [1] FacilityGID
facilityGroupMemberID [2] INTEGER
trunkGroupID [3] INTEGER
}

-- Indicates a logical identifier for the physical termination resource.

ResourceType ::= [45] IMPLICIT INTEGER (0..127)

```
-- This parameter is sent to indicate the type of resource to which a user is to be
-- connected.
--   Value         Meaning
--   0             Play Announcements
--   1             Play Announcements and Collect Digits
--   2             Text-to-Speech
--   3             Text-to-Speech and Collect Digits
--   4             Flex Parameter Block
--   5 - 126       Reserved for future capabilities
--   127           Correlation Block
```

ResponseCondition ::= ENUMERATED {
intermediateResponse(0),
lastResponse(1)
}

-- ResponseCondition is used to identify the reason why ServiceFilteringResponse operation is sent.
-- intermediateresponse identifies that service filtering is running and the interval time is expired and
-- a call is received, or that service filtering is running and the threshold value is reached.
-- lastResponse identifies that the duration time is expired and service filtering has been finished or
-- that the stop time is met and service filtering has been finished.

RouteIndex ::= [30] IMPLICIT OCTET STRING (SIZE(4))

```
--  RouteIndex Octet String:
--
--  | H | G | F | E | D | C | B | A |
--  |_______________|_______________|
--  |   2nd Digit   |   1st Digit   |
--  |_______________|_______________|
--  |   4th Digit   |   3rd Digit   |
--  |_______________|_______________|
--  |   6th Digit   |   5th Digit   |
--  |_______________|_______________|
--  |   8th Digit   |   7th Digit   |
--  |_______________|_______________|
--
--*  The range of each digit is 0-9
```

Sap ::= [81] IMPLICIT OCTET STRING (SIZE(1..10))

-- This parameter is sent to inform the SCP/Adjunct about operator services involved in a call, or
-- service-specific information such as a distinctive alerting pattern. When the originating access is an SS7
-- trunk and the received IAM contains the Sap parameter, the SSP shall map the Sap parameter received in
-- the IAM directly to the Sap parameter in the INAP message.

ScfID ::= OCTET STRING (SIZE (1..16))

-- defined by network operator.
-- Indicates the SCF identity.
-- Used to derive the INAP address of the SCF to establish a connection between a requesting FE
-- and the specified SCF.
-- When ScfID is used in an operation which may cross an internetwork boundary, its encoding must
-- be understood in both networks; this requires bilateral agreement on the encoding.
-- Refer to 3.5/Q.713 “calling party address” parameter for encoding.
-- It indicates the SCCP address e.g. Global Title of the SCF,.
-- Other encoding schemes are also possible as a network specific option..

SCIBillingChargingCharacteristics ::= OCTET STRING (SIZE (minSCIBillingChargingLength..
maxSCIBillingChargingLength))

-- This parameter indicates the billing and/or charging characteristics. Its content is network operator-specific.

SecondAlternateBillingIndicator ::= [46] IMPLICIT BillingIndicator

-- *This parameter is sent to provide the AMA call type and service feature identifier for the second alternate*
-- *trunk group.*

SecondAlternateCarrier ::= [47] IMPLICIT CarrierFormat
-- range - 4 digits

SecondAlternateTrunkGroup ::= [48] IMPLICIT OCTET STRING (SIZE(5))
-- Uses same format as defined for AlternateTrunkGroup

ServiceContext ::= [83] IMPLICIT INTEGER (0..32767)

-- *This parameter is sent to provide the identification of the context of a service (e.g., service identification,*
-- *feature identification).*

ServiceInteractionIndicators ::= OCTET STRING (SIZE (minServiceInteractionIndicatorsLength..
maxServiceInteractionIndicatorsLength))
-- Indicators which are exchanged between SSP and SCP to resolve interactions between IN based
-- services and network based services, respectively between different IN based services.
-- The contents are network-specific.
-- The following example is listed to illustrate the use of this parameter:
-- CallToBeDiverted Allowed/NotAllowed Indicator
-- If the CallToBeDiverted indicator is set to NotAllowed, the destination exchange shall not allow any
-- divertion on the subjected call. By this, each service can pass the applicable indicators to inform the
-- destination exchange of how specific services are to be handled.

ServiceProviderID ::= CHOICE {
Ocn
MsrID
}
Ocn ::= [87] IMPLICIT IA5String (SIZE (4..8))
MsrID ::= [106] IMPLICIT INDigits -- range - 10 digits

-- *This parameter identifies the Message Storage and Retrieval System.*
-- *This is a placeholder. There are two parameters associated with the ServiceProviderID placeholder. The Ocn*
-- *parameter is sent to support surveillance functionality. When the SCP requests the SSP to update Message*
-- *Waiting Indicator (MWI) information, the MsrID parameter is sent to identify the Message Storage and*
-- *Retrieval (MSR) system to the SSP.*

SecurityEnvelope ::= [85] IMPLICIT OCTET STRING (SIZE(3..75))

SFBillingChargingCharacteristics ::= OCTET STRING (SIZE (minSFBillingChargingLength..
maxSFBillingChargingLength))
-- This parameter indicates the billing and/or charging characteristics for filtered calls.
-- Its content is network operator-specific

Spid ::= [49] IMPLICIT OCTET STRING (SIZE(3..20))
-- This parameter includes a character string with 3-20 IA5 characters,
-- the last 2 are required to be numeric in the range 00-62.

SSP ::= IA5String (SIZE(1..8)) -- References office-wide triggers

```
SSPUserResource ::= [103] IMPLICIT SEQUENCE {
-- A field of the AdministrableObject parameter.
        sSPUserResourceID       [1]                 SSPUserResourceID,
        mwi                     Mwi                 OPTIONAL,
        oNoAnswerTimer          OnoAnswerTimer      OPTIONAL,
        tNoAnswerTimer          TnoAnswerTimer      OPTIONAL
        }

SSPUserResourceID ::= CHOICE {
        dn                      [1] IMPLICIT Dn
--   reserved or not supported  [2]
--   reserved or not supported  [3]
        trunkGroupID            [4] IMPLICIT TrunkGroupID
--   reserved or not supported  [5]
--   reserved or not supported  [6]
        }

StatusCause ::= [66] IMPLICIT ENUMERATED {
        StatusMatch             (0),
        timeOut                 (1),
        error                   (2)
        }

STRConnection ::= [96] IMPLICIT BOOLEAN

--   This parameter is sent to indicate whether or not the event being reported resulted from a
--   Send_To_Resource message.
--   TRUE = reported event caused by Send_To_Resource
--   FALSE = reported event not caused by Send_To_Resource

StrParameterBlock ::= [50] CHOICE {
-- This parameter is sent to provide any information that an SSP or IP needs to utilize for the
-- capability specified by the resourceType parameter. Therefore, it can take one of a number of
-- different codings depending on the capability which is to be utilized. Each resource type value
-- implies its own parameter block and its own coding.
        announcementBlock       [0] IMPLICIT AnnouncementBlock
        announcementDigitBlock  [1] IMPLICIT AnnouncementDigitBlock
        flexParameterBlock      [2] IMPLICIT FlexParameterBlock
        correlationBlock        [3] IMPLICIT CorrelationBlock
        }

Tcm ::= [51] IMPLICIT INDigits
-- maximum 2 digits

TerminationIndicator ::= [67] IMPLICIT OCTET STRING (SIZE(1))

-- [  H  [   G      [  F    [   E  [     D      [      C       [   B  [   A   [
-- _____________________________________________________________________________
-- [spare[RerouteInd[DispInd[ExcInd[UnrelErrInd[BusyCauseInd[AnsInd[NMCntrl[
-- _____________________________________________________________________________
--
--   A  NM Control List Overflow Indication
--   _____________________________________
--   0  No
--   1  Yes
--
--   B  Answer Indication
```

```
--    ____________________
--    0  No
--    1  Yes
--
--    C  Busy Cause Indication
--    ________________________
--    0  No
--    1  Yes
--
--    D   Unrelated Error Indication
--    _____________________________
--    0   No
--    1  Yes
--
--    E  Exception Indication
--    _______________________
--    0  No
--    1  Yes
--
--    F  Display Text Provided Indication
--    ___________________________________
--    0  No
--    1  Yes
--
--    G  Reroute Indication
--    ____________________
--    0  No
--    1  Yes
```

TimeoutTimer ::= [139] IMPLICIT IntervalTime
-- This parameter is sent to indicate the value of the SSP timeout timer

TimerID ::= ENUMERATED {
tssf(0)
}
-- Indicates the timer to be reset.

TimerValue ::= Integer4
-- Indicates the timer value (in seconds).

TNoAnswerTimer ::= [99] IMPLICIT INTEGER (1..120)
-- This parameter is sent to indicate the value, in seconds, of the SSP terminating no answer timer.

Tone ::= SEQUENCE {
toneID [0] Integer4,
duration [1] Integer4 OPTIONAL
}
-- The duration specifies the length of the tone in seconds, value 0 indicates infinite duration.

TranslationType ::= [70] IMPLICIT INTEGER (0..255)
-- TranslationType:
-- 0 reserved
-- 1-31 Internetwork applications
-- 32-222 Spare
-- 223-252 Network-specific applications
-- 253 Internetwork applications
-- 254 Network-specific applications
-- 255 Reserved

TriggerCriteriaFlag ::= [68] IMPLICIT OCTET STRING (SIZE(2))

-- TriggerCriteriaFlag:

```
--    _________________________________________________
--    [ H  [ G  [  F  [ E  [ D  [ C  [ B  [ A  [
--    _________________________________________________
--    [                      Flag                      [
--    _________________________________________________
--    [              Trigger Criteria Type             [
--    _________________________________________________
--
-- Flag
--    HGFEDCBA
--    00000000     Do not activate trigger
--    00000001     Activate trigger
-- Trigger Criteria Type field of TriggerCriteriaFlag
--    HGFEDCBA
--    00000000   reserved or not supported
--    00000001   Vertical service code
--    00000010   Customized access
--    00000011   Customized intercom
--    00000100   NPA
--    00000101   NPA NXX
--    00000110   reserved or not supported
--    00000111   reserved or not supported
--    00001000   NPA NXX XXXX
--    00001001   reserved or not supported
--    00001010   reserved or not supported
--    00001011   reserved or not supported
--    00001100   N11
--    00001101   AFR
--    00001110   Shared interoffice trunk
--    00001111   Termination Attempt
--    00010000   Off hook immediate
--    00010001   Off hook delayed
--    00010010   Channel setup PRI
--    00010011   NPA N
--    00010100   NPA NX
--    00010101   NPA NXXX
--    00010110   NPA NXXXX
--    00010111   NPA NXXXXX
--    00011000   reserved or not supported
--    00011001   T No Answer
--    00011010   TBusy
--    00011011   0 Called Party Busy
--    00011100   reserved or not supported
--    00011101   0 No Answer
--    00011110   reserved or not supported
--    00110010   facilitySelectedAndAvailable
```

TriggerCriteriaType ::= [52] IMPLICIT ENUMERATED {

FeatureActivator	**(0),**
verticalServiceCode	**(1),**
customizedAccess	**(2),**
customizedIntercom	**(3),**
npa	**(4),**
npaNXX	**(5),**
nxx	**(6),**
nxxXXXX	**(7),**
npaNXXXXXX	**(8),**
countryCodeNPANXXXXXX	**(9),**
carrierAccess	**(10),**
prefixes	**(11),**
n11	**(12),**
aFR	**(13),**
sharedIOTrunk	**(14),**
terminationAttempt	**(15),**
offHookImmediate	**(16),**

```
        offHookDelay                    (17),
        channelSetupPRI                 (18),
        npaN                            (19),
        npaNX                           (20),
        npaNXXX                         (21),
        npaNXXXX                        (22),
        npaNXXXXX                       (23),
        networkBusy                     (24),
        tNoAnswer                       (25),
        tBusy                           (26),
        oCalledPartyBusy                (27),
        specificFeatureCode             (28),
        oNoAnswer                       (29),
        priNetworkServices              (30),
        oSwitchHookFlashImmediate       (31),
        oFeatureActivator               (32),
        oSwitchHookFlashSpecifiedCode   (33),
        tSwitchHookFlashImmediate       (34),
        tFeatureActivator               (35),
        tSwitchHookFlashSpecifiedCode   (36),
        numberPortability               (37),
        onePlus                         (38),
        specifiedCarrier                (39),
        international                   (40),
        zeroPlus                        (41),
        zeroMinus                       (42),
        reserved                        (43),
        reserved                        (44),
        reserved                        (45),
        termResourceAvailable           (46),
        dedicatedTrunkGroup             (49),
        facilitySelectedAndAvailable    (50)
        }
-- For dedicatedTrunkGroup, see T1.628 on Emergency Calling Service description and usage.

TriggerItemAssignment ::= [102] IMPLICIT SEQUENCE {
-- A field of the AdministrableObject parameter.
        sSPUserResourceID           [1] SSPUserResourceID,
        triggerItemID               [2] IMPLICIT TriggerItemID,
        activationStateCode         [3] IMPLICIT ActivationStateCode        OPTIONAL
        }

TriggerItemID ::= SEQUENCE {
-- A sub-field of the triggerItemAssignment field of the AdministrableObject parameter.
        dPNumber                    [1] IMPLICIT DPNumber
        TriggerItemSubnumber        [2] IMPLICIT TriggerItemSubnumber
        }

TriggerItemSubnumber ::= IA5String (SIZE(1..8))
-- Identifies triggers and events within DPs.
```

```
DPNumber ::= INTEGER (1..33)
-- Identifies a Detection Point within the AIN Call Model.

TrunkGroupID ::= INTEGER (0..9999)

UnavailableNetworkResource ::= ENUMERATED {
        unavailableResources(0),
        componentFailure(1),
        basicCallProcessingException(2),
        resourceStatusFailure(3),
        endUserFailure(4)
        }
-- Indicates the network resource that failed.

UninterAnnounceBlock ::= SEQUENCE SIZE (1..10) OF AnnounceElement

UserID ::= [53] CHOICE {
        dn                      [1] IMPLICIT Dn
        isdnI                   [2] IMPLICIT SEQUENCE {
        spid                    Spid,
        dn                      Dn}                             -- BRI (ISDNI)
        trunkGroupID            [5] IMPLICIT TrunkGroupID       -- a type of FacilityGID
        privateFacilityGID      [6] IMPLICIT PrivateFacilityGID
        adsiCpeId               [7] IMPLICIT AdsiCpeID
        ssp                     [10] IMPLICIT Ssp
        dialingPlan             [11] IMPLICIT DialingPlan
        }

VariablePart ::= CHOICE {
        integer                 [0] Integer4
        number                  [1] Digits                      -- Generic digits
        time                    [2] OCTET STRING (SIZE(2))      -- HH:MM, BCD coded
        date                    [3] OCTET STRING (SIZE(3))      -- YYMMDD, BCD coded
        price                   [4] OCTET STRING (SIZE(4))
        }
-- Indicates the variable part of the message.
-- BCD coded variable parts are encoded as described in the examples below.
-- For example, time = 12:15 would be encoded as:
--      Bits  HGFE      DCBA
--      leading octet   2 1
--      5      1
-- date = 1993 September 30th would be encoded as:
--      Bits  HGFE      DCBA
--      leading octet   3 9
--      9      0
--      0      3

VerticalServiceCode ::= [54] IMPLICIT INDigits
-- Maximum 10 characters; the 1st character may be a * or digit.
-- The remaining characters are always digits.
-- The Definition of range of constants Follows
-- minAChBillingChargingLength          INTEGER ::=     -- network-specific
-- maxAChBillingChargingLength          INTEGER ::=     -- network-specific
-- minCallResultLength                  INTEGER ::=     -- network-specific
```

```
-- maxCallResultLength                          INTEGER ::=          -- network-specific
-- minCauseLength                               INTEGER ::= 2
-- maxCauseLength                               INTEGER ::=          -- network-specific
-- minDisplayInformationLength                  INTEGER ::=          -- network-specific
-- maxDisplayInformationLength                  INTEGER ::=          -- network-specific
-- minEventSpecificInformationChargingLength    INTEGER ::=          -- network-specific
-- maxEventSpecificInformationChargingLength    INTEGER ::=          -- network-specific
-- minEventTypeChargingLength                   INTEGER ::=          -- network-specific
-- maxEventTypeChargingLength                   INTEGER ::=          -- network-specific
-- minFCIBillingChargingLength                  INTEGER ::=          -- network-specific
-- maxFCIBillingChargingLength                  INTEGER ::=          -- network-specific
-- minLocationNumberLength                      INTEGER ::=          -- network-specific
-- maxLocationNumberLength                      INTEGER ::=          -- network-specific
-- minSCIBillingChargingLength                  INTEGER ::=          -- network-specific
-- maxSCIBillingChargingLength                  INTEGER ::=          -- network-specific
-- minServiceInteractionIndicatorsLength        INTEGER ::=          -- network-specific
-- maxServiceInteractionIndicatorsLength        INTEGER ::=          -- network-specific
-- minSFBillingChargingLength                   INTEGER ::=          -- network-specific
-- maxSFBillingChargingLength                   INTEGER ::=          -- network-specific

END
```

7.3.1.4 SSF-SCF packages, contracts, and application contexts

The **ssf-scf-contract** expresses the form of the service in which the SSF, a ROS-object of class **ssf**, initiates the contract. A ROS-object of class **scf** responds to this contract.

```
ssf-scf-contract CONTRACT ::= {
-- dialogue initiated by the SSF
        INITIATOR CONSUMER OF
        {sSFCallRelatedPackage|
        sSFNonCallRelatedPackage|
        abnormalPackage
        }
        RESPONDER CONSUMER OF
        {sCFCallRelatedPackage|
        sCFNonCallRelatedPackage|
        abnormalPackage
        }
        ID id-contract-ssf-scf
        }
```

The operation packages below are defined as information objects of class OPERATION-PACKAGE. The operations of these packages are defined in 7.3.1.1.

```
sSFCallRelatedPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES
        {analyzedInformation|
        callInfoFromResource|
        close|
        collectedInformation|
        facilitySelectedAndAvailable|
        oAnswer|
        oCalledPartyBusy|
        oDisconnect|
```

```
            oDTMFEntered|
            oNoAnswer|
            originationAttempt|
            oSuspended|
            oTermSeized|
            resourceClear|
            tAnswer|
            tBusy|
            terminationAttempt|
            timeout|
            tNoAnswer
            }
            ID id-package-sSFCallRelatedPackage
            }

sSFNonCallRelatedPackage OPERATION-PACKAGE ::= {
            CONSUMER INVOKES
            {monitorSuccess|
            nCAData|
            nCARequest|
            statusReported|
            terminationNotification
            }
            ID id-package-sSFNonCallRelatedPackage
            }

abnormalPackage OPERATION-PACKAGE ::= {
            CONSUMER INVOKES
            {reportError
            }
            ID id-package-abnormalPackage
            }

sCFCallRelatedPackage OPERATION-PACKAGE ::= {
            CONSUMER INVOKES
            {analyzeRoute|
            authorizeTermination|
            cancelResourceEvent|
            collectInformation|
            continue|
            forwardCall|
            offerCall|
            releaseCall|
            routeSelectFailure|
            sendToResource
            }
            ID id-package-sCFCallRelatedPackage
            }

sCFNonCallRelatedPackage OPERATION-PACKAGE ::= {
            CONSUMER INVOKES
            {acg|
```

```
        monitorForChange|
        nCAData|
        nCARequest|
        requestReportBCMEvent|
        sendNotification|
        update|
        updateRequest
        }
        ID id-package-sCFNonCallRelatedPackage
        }
```

Abstract Syntax

The SSF-SCF INAP ASEs that realize the operation packages specified share a single abstract syntax, **ssf-scf-abstract-syntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
ssf-scf-abstract-syntax ABSTRACT-SYNTAX ::= {
        BASIC-SSF-SCF-PDUs
        IDENTIFIED BY                   id-basic-ssf-scf
        }

BASIC-SSF-SCF-PDUs ::= TCMessage {{SSF-SCF-Invokable},{SSF-SCF-Returnable}}

SSF-SCF-Invokable OPERATION ::= {
        acg|
        analyzedInformation|
        analyzeRoute|
        authorizeTermination|
        callInfoFromResource|
        cancelResourceEvent|
        close|
        collectedInformation|
        collectInformation|
        continue|
        facilitySelectedAndAvailable|
        forwardCall|
        monitorForChange|
        monitorSuccess|
        nCAData|
        nCARequest|
        oAnswer|
        oCalledPartyBusy|
        oDisconnect|
        oDTMFEntered|
        offerCall|
        oNoAnswer|
        originationAttempt|
        oSuspended|
        oTermSeized|
        releaseCall|
        reportError|
        requestReportBCMEvent|
```

```
            resourceClear|
            routeSelectFailure|
            sendNotification|
            sendToResource|
            statusReported|
            tAnswer|
            tBusy|
            terminationAttempt|
            terminationNotification|
            timeout|
            tNoAnswer|
            update|
            updateRequest
            }

SSF-SCF-Returnable OPERATION ::= {
            acg|
            analyzedInformation|
            analyzeRoute|
            authorizeTermination|
            callInfoFromResource|
            cancelResourceEvent|
            close|
            collectedInformation|
            collectInformation|
            continue|
            facilitySelectedAndAvailable|
            forwardCall|
            monitorForChange|
            monitorSuccess|
            nCAData|
            nCARequest|
            oAnswer|
            oCalledPartyBusy|
            oDisconnect|
            oDTMFEntered|
            offerCall|
            oNoAnswer|
            originationAttempt|
            oSuspended|
            oTermSeized|
            releaseCall|
            reportError|
            requestReportBCMEvent|
            resourceClear|
            routeSelectFailure|
            sendNotification|
            sendToResource|
            statusReported|
            tAnswer|
            tBusy|
            terminationAttempt|
```

terminationNotification|
timeout|
tNoAnswer|
update|
updateRequest
}

SSF-SCF Application Contexts

The **ssf-scf-contract** is realized by an application context **ssf-scf-ac**. This is specified as an information object of the class APPLICATION-CONTEXT.

```
ssf-scf-ac APPLICATION-CONTEXT ::= {
        CONTRACT                        ssf-scf-contract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax|
                                        ssf-scf-abstract-syntax
                                        }
        APPLICATION CONTEXT NAME        id-ac-ssf-scf
        }
```

7.3.1.5 SSF-SCF ASN.1 modules

```
ANSI-IN-SSF-SCF-pkgs-contracts-acs { iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0)
inapmodule(3) version1(0)}

DEFINITIONS ::=

BEGIN

-- This module describes the operation packages, contracts, and application context.

IMPORTS
        ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, OPERATION
        FROM IN-Remote-Operations-Information-Objects {iso(1) memberbody(2) us(840) t1-667-
        2002(10069) modules(0) informationObjects(5) version1(0)}

        TCMessage
        FROM TCAPMessages {ccitt recommendation q773 modules(2) messages(1) version3(3)}

        APPLICATION-CONTEXT, dialogue-abstract-syntax
        FROM TC-Notation-Extensions {ccitt recommendation q775 modules(2) notation-extensions(4)
        version1(1)}

        acg,
        analyzedInformation,
        analyzeRoute,
        authorizeTermination,
        callInfoFromResource,
        cancelResourceEvent,
        close,
```

```
        collectedInformation,
        collectInformation,
        continue,
        facilitySelectedAndAvailable,
        forwardCall,
        monitorForChange,
        monitorSuccess,
        nCAData,
        nCARequest,
        oAnswer,
        oCalledPartyBusy,
        oDisconnect,
        oDTMFEntered,
        offerCall,
        oNoAnswer,
        originationAttempt,
        oSuspended,
        oTermSeized,
        releaseCall,
        reportError,
        requestReportBCMEvent,
        resourceClear,
        routeSelectFailure,
        sendNotification,
        sendToResource,
        statusReported,
        tAnswer,
        tBusy,
        terminationAttempt,
        terminationNotification,
        timeout,
        tNoAnswer,
        update,
        updateRequest
        FROM ANSI-IN-Operations { iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inap(0)
        version1(0)}

-- Application Context

ssf-scf-ac APPLICATION-CONTEXT ::= {
        CONTRACT                        ssf-scf-contract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax|
                                        ssf-scf-abstract-syntax
                                        }
        APPLICATION CONTEXT NAME        id-ac-ssf-scf
        }

-- Contracts

ssf-scf-contract CONTRACT ::= {
-- dialogue initiated by the SSF
```

```
        INITIATOR CONSUMER OF
        {sSFCallRelatedPackage|
        sSFNonCallRelatedPackage|
        abnormalPackage
        }
        RESPONDER CONSUMER OF
        {sCFCallRelatedPackage|
        sCFNonCallRelatedPackage|
        abnormalPackage
        }
        ID id-contract-ssf-scf
        }

-- Packages

sSFCallRelatedPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES
        {analyzedInformation|
        callInfoFromResource|
        close|
        collectedInformation|
        facilitySelectedAndAvailable|
        oAnswer|
        oCalledPartyBusy|
        oDisconnect|
        oDTMFEntered|
        oNoAnswer|
        originationAttempt|
        oSuspended|
        oTermSeized|
        resourceClear|
        tAnswer|
        tBusy|
        terminationAttempt|
        timeout|
        tNoAnswer
        }
        ID id-package-sSFCallRelatedPackage
        }

sSFNonCallRelatedPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES
        {monitorSuccess|
        nCAData|
        nCARequest|
        statusReported|
        terminationNotification
        }
        ID id-package-sSFNonCallRelatedPackage
        }

abnormalPackage OPERATION-PACKAGE ::= {
```

```
            CONSUMER INVOKES
            {reportError
            }
            ID id-package-abnormalPackage
            }

sCFCallRelatedPackage OPERATION-PACKAGE ::= {
            CONSUMER INVOKES
            {analyzeRoute|
            authorizeTermination|
            cancelResourceEvent|
            collectInformation|
            continue|
            forwardCall|
            offerCall|
            releaseCall|
            routeSelectFailure|
            sendToResource
            }
            ID id-package-sCFCallRelatedPackage
            }

sCFNonCallRelatedPackage OPERATION-PACKAGE ::= {
            CONSUMER INVOKES
            {acg|
            monitorForChange|
            nCAData|
            nCARequest|
            requestReportBCMEvent|
            sendNotification|
            update|
            updateRequest
            }
            ID id-package-sCFNonCallRelatedPackage
            }

-- Abstract Syntax

ssf-scf-abstract-syntax ABSTRACT-SYNTAX ::= {
            BASIC-SSF-SCF-PDUs
            IDENTIFIED BY id-basic-ssf-scf
            }

BASIC-SSF-SCF-PDUs ::= TCMessage {{SSF-SCF-Invokable},{SSF-SCF-Returnable}}

SSF-SCF-Invokable OPERATION ::= {
            acg|
            analyzedInformation|
            analyzeRoute|
            authorizeTermination|
            callInfoFromResource|
            cancelResourceEvent|
```

```
            close|
            collectedInformation|
            collectInformation|
            continue|
            facilitySelectedAndAvailable|
            forwardCall|
            monitorForChange|
            monitorSuccess|
            nCAData|
            nCARequest|
            oAnswer|
            oCalledPartyBusy|
            oDisconnect|
            oDTMFEntered|
            offerCall|
            oNoAnswer|
            originationAttempt|
            oSuspended|
            oTermSeized|
            releaseCall|
            reportError|
            requestReportBCMEvent|
            resourceClear|
            routeSelectFailure|
            sendNotification|
            sendToResource|
            statusReported|
            tAnswer|
            tBusy|
            terminationAttempt|
            terminationNotification|
            timeout|
            tNoAnswer|
            update|
            updateRequest
            }

SSF-SCF-Returnable OPERATION ::= {
            acg|
            analyzedInformation|
            analyzeRoute|
            authorizeTermination|
            callInfoFromResource|
            cancelResourceEvent|
            close|
            collectedInformation|
            collectInformation|
            continue|
            facilitySelectedAndAvailable|
            forwardCall|
            monitorForChange|
            monitorSuccess|
```

```
        nCAData|
        nCARequest|
        oAnswer|
        oCalledPartyBusy|
        oDisconnect|
        oDTMFEntered|
        offerCall|
        oNoAnswer|
        originationAttempt|
        oSuspended|
        oTermSeized|
        releaseCall|
        reportError|
        requestReportBCMEvent|
        resourceClear|
        routeSelectFailure|
        sendNotification|
        sendToResource|
        statusReported|
        tAnswer|
        tBusy|
        terminationAttempt|
        terminationNotification|
        timeout|
        tNoAnswer|
        update|
        updateRequest
        }
END

ANSI-IN-ITU-SSF-SCF-pkgs-contracts-acs { iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0)
inapmodule2(4) version1(0}
DEFINITIONS ::=
BEGIN

-- This module describes the operation packages, contracts, and application context.

IMPORTS
        ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, OPERATION
        FROM IN-Remote-Operations-Information-Objects {iso(1) memberbody(2) us(840) t1-667-
         2002(10069) modules(0) informationObjects(5) version1(0)}

        TCMessage
        FROM TCAPMessages {ccitt recommendation q773 modules(2) messages(1) version3(3)}

        APPLICATION-CONTEXT, dialogue-abstract-syntax
        FROM TC-Notation-Extensions {ccitt recommendation q775 modules(2) notation-extensions(4)
        version1(1)}

        activateServiceFiltering,
        activityTest,
        applyCharging,
```

```
            applyChargingReport,
            callInformationReport,
            callInformationRequest,
            eventNotificationCharging,
            furnishChargingInformation,
            oMidCall,
            requestNotificationChargingEvent,
            resetTimer,
            sendChargingInformation,
            serviceFilteringResponse,
            tMidCall
            FROM ANSI-IN-Operations { iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inap(0)
            version1(0)}

-- Application Context

ssf-scf-ac2 APPLICATION-CONTEXT ::= {
            CONTRACT                    ssf-scf-contract2
            DIALOGUE MODE               structured
            TERMINATION                 basic
            ABSTRACT SYNTAXES           {dialogue-abstract-syntax|
                                        ssf-scf-abstract-syntax
                                        }
            APPLICATION CONTEXT NAME    id-ac2-ssf-scf
            }

-- Contracts

ssf-scf-contract2 CONTRACT ::= {
-- dialogue initiated by the SSF
            INITIATOR CONSUMER OF
            {activateServiceFiltering|
            activityTest|
            applyCharging|
            applyChargingReport|
            callInformationReport|
            callInformationRequest|
            eventNotificationCharging|
            furnishChargingInformation|
            oMidCall|
            requestNotificationChargingEvent|
            resetTimer|
            sendChargingInformation|
            serviceFilteringResponse|
            tMidCall
            }
            RESPONDER CONSUMER OF
            {activateServiceFiltering|
            activityTest|
            applyCharging|
            applyChargingReport|
            callInformationReport|
            callInformationRequest|
```

```
            eventNotificationCharging|
            furnishChargingInformation|
            oMidCall|
            requestNotificationChargingEvent|
            resetTimer|
            sendChargingInformation|
            serviceFilteringResponse|
            tMidCall
            }
            ID id-contract2-ssf-scf
            }

END
```

7.3.1.6 SCF-SRF ASN.1 module

```
ANSI-IN-ITU-SCF-SRF-pkgs-contracts-acs { iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0)
inapmodule3(7) version1(0}

DEFINITIONS ::=

BEGIN

-- This module describes the operation packages, contracts, and application context.

IMPORTS
            ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, OPERATION
            FROM IN-Remote-Operations-Information-Objects {iso(1) memberbody(2) us(840) t1-667-
             2002(10069) modules(0) informationObjects(5) version1(0)}

            TCMessage
            FROM TCAPMessages {ccitt recommendation q773 modules(2) messages(1) version3(3)}

            APPLICATION-CONTEXT, dialogue-abstract-syntax
            FROM TC-Notation-Extensions {ccitt recommendation q775 modules(2) notation-extensions(4)
            version1(1)}

            assistRequestInstructions,
            instructionsToSRF
            FROM ANSI-IN-Operations {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) inap(0)
            version1(0)}

-- Application Context

scf-srf-ac APPLICATION-CONTEXT ::= {
            CONTRACT                    scf-srf-contract
            DIALOGUE MODE               structured
            TERMINATION                 basic
            ABSTRACT SYNTAXES           {dialogue-abstract-syntax|
                                        scf-srf-abstract-syntax
                                        }
            APPLICATION CONTEXT NAME    id-ac-scf-srf
            }
```

```
-- Contracts

scf-srf-contract CONTRACT ::= {
-- dialogue initiated by the SRF
        INITIATOR CONSUMER OF
        {assistRequestInstructions|
        callInfoFromResource
        }
        RESPONDER CONSUMER OF
        {instructionsToSRF|
        callInfoFromResourceRR
        }
        ID id-contract-scf-srf
        }

END
```

7.4 Semantics

7.4.1 Definition of procedures and entities

7.4.1.1 SSF application entity procedures

Refer to ITU-T Recommendation Q.1218, “Interface Recommendation for intelligent network CS-1”, for details.

7.4.1.2 SCF application entity procedures

Refer to ITU-T Recommendation Q.1218, “Interface Recommendation for intelligent network CS-1”, for details.

7.4.1.3 SRF application entity procedures

Refer to ITU-T Recommendation Q.1218, “Interface Recommendation for intelligent network CS-1”, for details.

7.4.2 Error procedures

This clause defines the generic error procedures for the T1 IN INAP. The error procedure descriptions have been divided in two clauses: 7.4.2.1 listing the errors related to INAP operations and 7.4.2.2 listing the errors related to error conditions in the different FEs that are not directly related to the INAP operations.

7.4.2.1 Operation related error procedures

The following clauses define the generic error handling for the operation related errors. The errors are defined as operation errors in 7.3.1.2. The TCAP services, which are used for reporting operation errors, are described in 7.4.7.

Errors that have a specific procedure for an operation are described in 7.4.3 with the detailed procedure of the related operation.

All errors, which can be detected by the ASN.1 decoder, already may be detected during the decoding of the TCAP message and indicated by the TC error indication “MistypedParameter” in the Reject component.

7.4.2.1.1 Application Errors

Application Errors are errors that are associated with an APDU that violate the normal procedures associated with the APDU and are of non-protocol nature.

An SSF/SRF shall detect application errors in messages received from an SCF; and an SCF shall detect application errors in messages received from an SSF/SRF.

7.4.2.1.1.1 Fatal Application Errors

Fatal application errors are error conditions that cause an APDU not to be processed and may cause a transaction to be closed. These errors are not reported if the message was received in a Unidirectional Package.

The following errors are Fatal Application Errors.

7.4.2.1.1.1.1 Response Message Timer Expired

This error occurs when an SSF (SCF) sends a message and the response message timer expires before the timer is canceled.

If an SSF (SCF) detects a Response Message Timer Expired Application Error while waiting for a response, then it shall:

– provide final treatment or default routing;

– report the error.

7.4.2.1.1.1.2 Unexpected Message

An SSF (SCF) detects an Unexpected Message Application Error if it receives:

– a message in an open transaction and the associated timer had not been set;

– a message that is not allowed in the context of a given transaction;

– a message that is not allowed given the current state of resources;

– multiple components that are not allowed to be sent together.

If an SSF (SCF) detects an Unexpected Message Application Error in a call-related message then it shall:

– · discard the component received in error;

– · report the error;

– · provide final treatment or default routing;

– · process any other components in the message.

7.4.2.1.1.1.3 Unexpected Message Sequence

An SSF (SCF) detects an Unexpected Message Sequence Application Error if it receives:

– a TCAP package that contains a call-related message, but that is not the first message in the package; or

– a TCAP Conversation package and a required message are missing from the conversation package.

If an SSF (SCF) detects an Unexpected Message Sequence Application Error in a call-related message then it shall:

– discard the component received in error;

– report the error;

– provide final treatment or default routing;

– process any other components in the message.

7.4.2.1.1.1.4 Unexpected Parameter Sequence

An SSF (SCF) detects a fatal Unexpected Parameter Sequence Application Error if it receives a message and after dropping any unrecognized parameter, the remaining parameters do not follow any known ANSI Standard ASN.1 sequence description.

If an SSF (SCF) detects an Unexpected Parameter Sequence Application Error in a call-related message then it shall:

– discard the component received in error;

– report the error;

– provide final treatment or default routing;

– process any other components in the message.

7.4.2.1.1.1.5 Erroneous Data Value

An SSF (SCF) detects a fatal Erroneous Data Value Application Error if:

– a field within any parameter contained in the received message has an incorrect value and

– the operation cannot be performed if the parameter in error is dropped.

If an SSF (SCF) detects a fatal Erroneous Data Value Application Error in a call-related message then it shall:

– discard the component received in error;

– report the error;

– provide final treatment or default routing;

– process any other components in the message.

7.4.2.1.1.1.6 Unexpected Communication

An SSF (SCF) detects a fatal Unexpected Communication Application Error if it receives:

– a message in a Response Package which should have been received in a Conversation Package;

– a TCAP component is different from the one described for that message; or

– a Query Without Permission or a Conversation Without Permission Package.

If an SSF (SCF) detects a fatal Unexpected Communication Application Error in a call-related message then it shall:

– · discard the component received in error;

– · report the error;

– · provide final treatment or default routing;

– · process any other components in the message.

7.4.2.1.1.1.7 Missing Conditional Parameter

An SSF (SCF) detects a fatal Missing Conditional Parameter Application Error if it receives a message that cannot be processed because of a missing conditional parameter.

If an SSF (SCF) detects a fatal Missing Conditional Parameter Application Error then it shall:

– discard the component received in error;

– report the error;

– provide final treatment fault handling procedures;

– process any other components in the message.

7.4.2.1.1.2 Non-fatal Application Errors

Non-fatal errors are error conditions that do not interfere with the completion of an operation.

The following errors are Non-fatal Application Errors.

7.4.2.1.1.2.1 Erroneous Data Value

An SSF (SCF) detects a non-fatal Erroneous Data Value Application Error if:

– a field within any parameter contained in the received message has an incorrect value and

– it is determined in real time that the parameter in error is not required to perform the operation.

If an SSF (SCF) detects a non fatal Erroneous Data Value Application Error then it shall:

– drop the field received in error and

– continue to process the remaining message, from the next field of the parameter in question.

7.4.2.1.1.2.2 Unexpected Communication

An SSF (SCF) detects a non-fatal Unexpected Communication Application Error if it receives a message in a Conversation Package that should have been received in a Response Package.

If an SSF (SCF) detects a non-fatal Unexpected Communication Application Error and no other errors are associated with that TCAP message, then it shall:

– process the SCF message and

– close the transaction.

7.4.2.1.1.2.3 Missing Conditional Parameter

An SSF (SCF) detects a non-fatal Missing Conditional Parameter Application Error if it receives a message with a missing conditional parameter,

– which is not required to perform the operation or

– a default value can be provided.

If an SSF (SCF) detects a non fatal Missing Conditional Parameter Application Error then it shall continue to process the message.

7.4.2.1.1.2.4 Unexpected Parameter

An SSF (SCF) detects a non-fatal Unexpected Parameter Application Error if it receives a message:

– that contains a valid parameter, and

– it is determined in real time that this parameter is not required to perform the operation.

If an SSF (SCF) detects a non fatal Unexpected Parameter Application Error then it shall

– drop the unexpected parameter, and

– continue to process the message.

7.4.2.1.2 Failures

A failure occurs when a hardware or software resource is unable to provide functionality that is necessary for the completion of an operation. The *FailureCause* parameter in the Failure_Report message indicates the cause of failure. Either final treatment or default routing should be applied to the call (network operator-specific).

7.4.2.1.3 MissingCustomerRecord

7.4.2.1.3.1 Error description

The Service Logic Program could not be found in the SCF, because the required customer record does not exist, or the requested Service Logic Program Instance, indicated by the correlationID does not exist any more. These two cases should be distinguished as two different error situations, because the error procedure shows that the occurrence of the MissingCustomerRecord error is reported to the maintenance function, but the report to the maintenance function for the occurrence of the former case should be optional because it occurs not only in extraordinary situations but in ordinary situations. For example, the former may occur when the end user dials a missing free-phone number.

7.4.2.1.3.2 Operations SSF->SCF

The operations on this interface are:

OMidCall

TMidCall

The CCF routes the call if necessary (e.g., default routing to a terminating announcement).

If the SCSM detects that, the required Service Logic Program does not exist then the Error parameter MissingCustomerRecord is used to inform the invoking entity of this situation. The Service Logic Program Instance may not exist anymore, or the Service Logic Program may have never existed at all (i.e., the customer record in the SCF does not exist, e.g., in case of TDPs a Service Logic Program is attempted to be invoked). The maintenance functions are informed (however, it is optional for the TDP operation case).

7.4.2.1.4 MissingParameter

7.4.2.1.4.1 Error description

There is an Error in the received Operation argument. The responding entity cannot start to process the requested Operation because the argument is incorrect: a mandatory parameter (the application shall always return this error in case it is not detected by the ASN.1 decoder) or an expected optional parameter that is essential for the application is not included in the Operation argument.

7.4.2.1.4.2 Operations SCF->SSF

The operations on this interface are:

ActivateServiceFiltering

ApplyCharging

CallInformationRequest

FurnishChargingInformation

RequestNotificationChargingEvent

ResetTimer

SendChargingInformation

The Service Logic and maintenance functions are informed. Further treatment of the call is dependent on Service Logic.

If the SSF FSM detects the error in the received operation then the Error parameter is returned to inform the SCF of this situation.

7.4.2.1.4.3 Operations SSF->SCF

The operations on this interface are:

ApplyChargingReport

OMidCall

TMidCall

After receiving this Error, the SSF FSM returns to the state Idle. The CCF routes the call if necessary (default routing to a terminating announcement). If the call is already established (i.e., mid-call trigger or ApplyChargingReport), the CCF may maintain the call or disconnect it. The choice between these two options is network operator-specific. In case of an assisting SSF, the temporary connection is released by the assisting SSF.

If the SCSM detects the erroneous situation then the Error parameter is used to inform the SSF of this situation. The Service Logic and maintenance functions are informed.

7.4.2.1.5 ParameterOutOfRange

7.4.2.1.5.1 Error description

The responding entity cannot start the processing of the requested Operation because an Error in a parameter of the Operation argument is detected: a parameter value is out of range. This error is applied for the following two cases (when the error is determined by the application):

(1) For the parameter which type is defined with the range of its size, such as INTEGER (x..y), SEQUENCE SIZE (x..y) OF Type. This error is applied when the parameter value is z or the parameter size is z where z<x or z>y.

(2) For the parameter which type is defined as list of ENUMERATED value, the ParameterOutOfRange error is applied when the parameter value is not equal to any of the ENUMERATED values in the list.

7.4.2.1.5.2 Operations SCF->SSF

The operations on this interface are:

ActivateServiceFiltering

ApplyCharging

CallInformationRequest

RequestNotificationChargingEvent

ResetTimer

SendChargingInformation

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.5.3 Operations SSF->SCF

The operations on this interface are:

ApplyChargingReport

OMidCall

TMidCall

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.6 RequestedInfoError

7.4.2.1.6.1 General description

– Error description

The RequestedInfoError is an immediate response to the CallInformationRequest operation, indicating that the requested information is not known to the SSF or is not available. RequestedInfoError is used when a specific SSF/CCF cannot offer the information specified with RequestedInformationType but there exists other SSF/CCF that can offer the information.

– Argument description

The parameter is RequestedInfoError.

7.4.2.1.6.2 Operations SCF->SSF

The operations on this interface are:

CallInformationRequest

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.7 SystemFailure

7.4.2.1.7.1 General description

- Error description

This error is returned by a physical entity if it was not able to fulfill a specific task as requested by an operation, and recovery is not expected to be completed within the current call instance.

- Argument description

The parameter is UnavailableNetworkResource.

7.4.2.1.7.2 Operations SCF->SSF

The operations on this interface are:

ActivateServiceFiltering

ApplyCharging

CallInformationRequest

RequestNotificationChargingEvent

SendChargingInformation

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.7.3 Operations SSF->SCF

The operations on this interface are:

ApplyChargingReport

OMidCall

TMidCall

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.8 TaskRefused

7.4.2.1.8.1 General introduction

- Error description

This Error is returned by a physical entity if it was not able to fulfill a specific task as requested by an operation, and recovery is expected to be completed within the current call instance.

- Argument description

The parameter is TaskRefused.

7.4.2.1.8.2 Operations SCF->SSF

The operations on this interface are:

ActivateServiceFiltering

ApplyCharging

CallInformationRequest

FurnishChargingInformation

RequestNotificationChargingEvent

ResetTimer

SendChargingInformation

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.8.3 Operations SSF->SCF

The operations on this interface are:

ApplyChargingReport

OMidCall

TMidCall

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.9 UnexpectedComponentSequence

7.4.2.1.9.1 Error description

The responding entity cannot start the processing of the requested operation because a SACF or MACF rule is violated, or the operation could not be processed in the current state of the FSM.

7.4.2.1.9.2 Operations SCF->SSF

The operations on this interface are:

ActivateServiceFiltering

ApplyCharging

CallInformationRequest

FurnishChargingInformation

RequestNotificationChargingEvent

RequestReportBCMEvent

ResetTimer

SendChargingInformation

In this case the SSF detects the erroneous situation, sends the UnexpectedComponentSequence error and remains in the same state. In the SCF, the Service Logic and maintenance functions are informed and the Service Logic decides about error treatment.

7.4.2.1.9.3 Operations SSF->SCF

The operations on this interface are:

ApplyChargingReport

OMidCall

TMidCall

In case the operation is sent by an “initiating” SSF in the context of an existing relationship, the SCF returns the error parameter. Service Logic and maintenance are informed. On receiving the error, the SSF moves to Idle.

7.4.2.1.10 UnexpectedDataValue

7.4.2.1.10.1 Error description

The responding entity cannot complete the processing of the requested Operation because a parameter has an unexpected data value.

Note that this error does not overlap with “ParameterOutOfRange”

Example: startTime DateAndTime ::= -- value indicating January 32 1999, 12:15:01

The responding entity does not expect this value and responds with “UnexpectedDataValue”.

7.4.2.1.10.2 Operations SCF->SSF

The operations on this interface are:

ActivateServiceFiltering

ApplyCharging

CallInformationRequest

FurnishChargingInformation

RequestNotificationChargingEvent

ResetTimer

SendChargingInformation

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.10.3 Operations SSF->SCF

The operations on this interface are:

ApplyChargingReport

OMidCall

TMidCall

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.11 UnexpectedParameter

7.4.2.1.11.1 Error description

There is an error in the received Operation argument. A valid but unexpected parameter was present in the Operation argument. The presence of this parameter is not consistent with the presence of the other parameters. The responding entity cannot start to process the Operation.

7.4.2.1.11.2 Operations SCF->SSF

The operations on this interface are:

ActivateServiceFiltering

ApplyCharging

CallInformationRequest

FurnishChargingInformation

RequestNotificationChargingEvent

ResetTimer

SendChargingInformation

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.11.3 Operations SSF->SCF

The operations on this interface are:

ApplyChargingReport

OMidCall

TMidCall

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.1.12 UnknownLegID

7.4.2.1.12.1 Error description

This error is used to indicate to the SCF that a specific leg, indicated by the LegID parameter value in the operation, is unknown to the SSF.

7.4.2.1.12.2 Operations SCF->SSF

The operations on this interface are:

SendChargingInformation

Refer to 7.4.2.1.4 MissingParameter for the appropriate error procedures.

7.4.2.2 Entity related error procedures

The following clauses define the error handling for the entity related errors. Since the error situations are not originated by the reception of an operation, the invoking entity is denoted here as the entity at which the error situation is detected. The responding entity is the entity that receives the error report.

The TCAP services used for reporting errors are described in 7.4.7.

7.4.2.2.1 Expiration of TSSF

7.4.2.2.1.1 Error description

A timeout occurred in the SSF on the response from the SCF.

7.4.2.2.1.2 Procedures SSF->SCF

The SSF FSM aborts the dialogue and moves to the Idle state, the CCF routes the call if necessary (e.g., default routing to a terminating announcement). The abort is reported to the maintenance functions.

The SCF releases all allocated resources and reports the abort to the maintenance functions, if the abort is received on an SSF dialogue. The SCF releases all resources related to the dialogue, reports the abort to the maintenance functions and returns to state preparing SSF instructions, if the abort is received on an assisting SSF dialogue.

7.4.3 Detailed operation procedures: SSF-SCF interface

7.4.3.1 Acg procedure

7.4.3.1.1 General description

This operation is used to regulate the flow of messages between the SSF and the SCF during periods of congestion. This operation is used to request the SSF to reduce the rate at which specific service requests are sent to the SCF.

7.4.3.1.2 Parameters

- controlType (*ControlCauseIndicator*):
  This parameter indicates the type of control and the number of digits to which the control is being applied.
- gapIndicatorsDuration (*GapDuration*):
  This parameter indicates the gap duration in seconds.
- gapIndicatorsGapInterval (*GapInterval*):
  This parameter indicates the gap interval in seconds.
- translationType (*TranslationType*):
  This parameter indicates the appropriate Global Title Translation function.
- globalTitleAddress (*GlobalTitleAddress*):
  This parameter indicates an address, such as dialed digits, which does not explicitly contain information that would allow routing in the signaling network.
- extensionParameter (*ExtensionParameter*):
  This parameter is sent to provide network-specific information.

7.4.3.1.3 Invoking entity (SCF)

7.4.3.1.3.1 Normal procedure

The Acg operation supports both manual and automatic outgoing ACG controls. SCF overload controls are automatic and exercised at the 6-digit (NPA-NXX) level. Manual code controls may be requested with 3, 6, and 7 to 10 digits. The automatic SCF overload controls are introduced and adjusted based on SCF congestion level. The manual controls are manual in the sense that the gapping activity, while still automatic, is manually initiated, independent of SCF load.

The outgoing controls are based on indirect routing of SCF queries. A traffic control item is identified by its Global Title Address (GTA) and Translation Type (TT), which are converted within the signaling network to the signaling point code of the destination SCF and Subsystem Number (SSN) for the application or application set of the SCF.

7.4.3.1.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.1.4 Responding entity (SSF)

7.4.3.1.4.1 Normal procedure

For the GlobalTitleAddress and TranslationType received, the SSF should apply the appropriate gapping as indicated by the gapIndicatorsDuration, gapIndicatorsGapInterval and controlType.

7.4.3.1.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.2 ActivateServiceFiltering procedure

7.4.3.2.1 General description

When receiving this operation, the SSF handles calls to destinations in a specified manner without request for instructions to the SCF. In the case of service filtering the SSF executes a specific service filtering algorithm. Alternatively, the transfer of service filtering results refer to the operation “ServiceFilteringResponse”.

7.4.3.2.2 Parameters

– filteredCallTreatment (*FilteredCallTreatment*):
  This parameter specifies how filtered calls are treated. It includes information about the announcement to be played, the charging approach, the number of counters used, and the release cause to be applied to filtered calls.

- sFBillingChargingCharacteristics:
  This parameter determines the charging to be applied for service filtering. Its content is network-specific.
- informationToSend:
  This parameter indicates an announcement, a tone, or display information to be sent to the calling party. At the end of information sending, the call shall be released.
  - inbandInfo:
    This parameter specifies the inband information to be sent.
    - messageID:
      This parameter indicates the message(s) to be sent, which can be one of the following:
      - elementaryMessageID:
        This parameter indicates a single announcement.
      - text:
        This parameter indicates text to be sent. The text shall be transformed to inband information (speech). This parameter consists of two subparameters: messageContent and attributes. The attributes of text may consist of items such as language.
      - elementaryMessageIDs:
        This parameter specifies a sequence of announcements.
      - variableMessage:
        This parameter specifies an announcement with one or more variable parts.
    - numberOfRepetitions:
      This parameter indicates the maximum number of times the message shall be sent to the end-user.
    - duration:
      This parameter indicates the maximum time duration in seconds that the message shall be played or repeated. ZERO indicates endless repetition.
    - interval:
      This parameter indicates the time interval in seconds between repetitions, i.e., the time between the end of the announcement and the start of the next repetition. This parameter can only be used when the number of repetitions is > 1.
  - tone:
    This parameter specifies a tone to be sent to the end-user.
    - toneID:
      This parameter indicates the tone to be sent.
    - duration:
      This parameter indicates the time duration in seconds of the tone to be sent. ZERO indicates infinite duration.
  - displayInformation:
    This parameter indicates a text string to be sent to the end-user. This information cannot be received by a PSTN end-user.
- maximumNumberOfCounters:
  This parameter provides the number of counters to be allocated as well as the number of destinations included in the service filtering, i.e., “maximumNumberOfCounters” subsequent destination addresses beginning with the destination address provided in “filteringCriteria” are used for service filtering. One counter is assigned to each of these destination addresses.

The number of counters may only be >1 if the “filteringCriteria” are of the type “addressAndService”.

- releaseCause:
  This parameter provides the cause value used for call release after the “informationToSend” (for example, announcement) has been sent to the calling party. If “releaseCause” is not present, the default value is the same as the ISUP value decimal 31 (normal unspecified).

– filteringCharacteristics (*FilteringCharacteristics*):
This parameter indicates the severity of the filtering and the point in time when the “ServiceFilteringResponse” shall be sent. It determines whether the “interval” or the “numberOfCalls” is used.

- interval:
  After expiration of the interval timer the next call to arrive causes following actions:
  - sending of a DP-specific operation,
  - sending of a “ServiceFilteringResponse”,
  - starting again the interval timer.
  - When filtering is started the first interval is started.
  - An interval of 0 indicates that all calls matching the filtering criteria will result in sending of a DP-specific operation and no filtering will be applied (i.e., no “ServiceFilteringResponse” will be sent).
  - An interval of -1 indicates that none of the calls matching the filtering criteria will either result in sending of a DP-specific operation or a “ServiceFilteringResponse” operation.
  - Other values indicate duration in seconds.
- numberOfCalls:
  The Nth call causes a DP-specific operation and an “ServiceFilteringResponse” operation sent to the SCF. This threshold value is met if the sum of all counters assigned to one service filtering entity is equal to “numberOfCalls”.
- A number of calls of 0 indicates that none of the calls matching the filtering criteria will result in sending of a DP-specific operation and a “ServiceFilteringResponse” operation.

– filteringTimeOut (*FilteringTimeOut*):
This parameter indicates the duration of the filtering. When the time expires, a “ServiceFilteringResponse” is sent to the SCF and service filtering is stopped. Two approaches are supported (duration or stopTime):

- duration:
  - If the duration time expires, then service filtering is stopped and the final report is sent to the SCF.
  - A duration of 0 indicates that service filtering is to be removed.
  - A duration of -1 indicates an infinite duration.
  - A duration of -2 indicates a network-specific duration.
  - Other values indicate duration in seconds.
- stopTime:
  When the “stopTime” is met then service filtering is stopped and the final report is sent to the SCF. If “stopTime” was already met, i.e., the value of the stopTime is less than the value of the actual time but the difference does not exceed the value equivalent to 50 years, then service filtering is immediately stopped and the actual counter values are reported to the SCF. This occurs in cases where the SCF wishes to explicitly stop a running service filtering.

– filteringCriteria (*FilteringCriteria*):
  This parameter specifies which calls are filtered based on “serviceKey”, “callingAddressValue”, “calledAddressValue” or “locationNumber”. It is a choice of “serviceKey” or “addressAndService”.

  – serviceKey:
    This parameter identifies unambiguously the requested IN service for which filtering should be applied.

  – addressAndService:
    This parameter identifies the IN service and dialed number for which filtering should be applied. The geographical area may also be identified (“callingAddressValue” and/or “locationNumber”).

    - calledAddressValue:
      This parameter contains the dialed number towards which filtering shall be applied. The complete *Called Party Number* shall be specified.

    - serviceKey:
      This parameter identifies unambiguously the requested IN service for which filtering should be applied.

    - callingAddressValue:
      This parameter contains the calling party number that identifies the calling party or geographical origin of the call for which filtering shall be applied.

    - locationNumber:
      This parameter identifies the geographical area from which the call to be filtered originates. It is used when “callingAddressValue” does not contain any information about the geographical location of the calling party.

– startTime (*DateAndTime*):
  This parameter defines when filtering is started. If “startTime” is not provided or was already met, the SSF starts filtering immediately.

– extensions (*SEQUENCE OF ExtensionField*):
  This parameter indicates an extension of an argument data type. Its content is network-specific.

7.4.3.2.3 Invoking entity (SCF)

7.4.3.2.3.1 Normal procedure

SCF precondition:

SLPI detects that service filtering has to be initiated at the SSF.

SCF postconditions:

1. SLPI starts an application timer to monitor the expected end of service filtering.
2. The SCME is in the state “Waiting For ServiceFilteringResponse”.

Sending the “ActivateServiceFiltering” operation causes a transition of the SCME from the state “Service Filtering Idle” to the state “Waiting For SSF Service Filtering Response”. The SCME remains in this state until the application timer in the SLPI expires. The SCME is informed by the SLPI about timer expiration. Then it moves to the state “Service Filtering Idle”.

If no errors occurred after receiving an “ActivateServiceFiltering” at the SSF an empty Return Result is sent to the SCF. That causes no state transition in the SCME.

To change the parameters of an existing service filtering entity the SCF has to send an "ActivateServiceFiltering" operation with the same "filtering Criteria". The second parameter set replaces the first one.

7.4.3.2.3.2 Error handling

Generic error handling for the operation related errors are described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.2.4 Responding entity (SSF)

7.4.3.2.4.1 Normal procedure

SSF preconditions:

None.

SSF postcondition:

The SSME-FSM is in the state "Non Call Associated Treatment".

If there is no already existing SSME-FSM for the "filteringCriteria" provided, then a new SSME-FSM is created. This SSME-FSM enters the state "Non-Call Associated Treatment" and initializes the service filtering for the specified IN calls. The parameters "filteredCallTreatment", "filteringCharacteristics", "filteringCriteria", "filteringTimeOut" and "startTime" are set as provided in the operation. A number of counters will be allocated and reset. In the case of the "startTime" that has not been met yet, the service filtering will be started at the specified point in time.

If the operation "ActivateServiceFiltering" addresses an already existing service filtering entity, the parameters "filteredCallTreatment", "filteringCharacteristics" "filteringTimeOut" and "startTime" are modified as provided in the operation. In the case that the addressed service filtering entity is active, the SSF reports the counter values to the SCF via the operation "ServiceFilteringResponse". The service filtering process is stopped if an already expired "stopTime" or "duration" equal to ZERO or a new not yet met "startTime" is provided. The SSF then proceeds as described for "ServiceFilteringResponse". In the case of the "startTime" that has not been met yet, the service filtering will be continued at the specified point in time.

If the service filtering proceeds then the SSME-FSM remains in the state "Non-Call Associated Treatment". Otherwise, the SSME-FSM moves to state "Idle Management".

When a call matches several active "filteringCriteria" it should be subject to filtering on the most specific criterion, i.e., the criterion with the longest "callingAddressValue" or "locationNumber", or alternatively the criterion with the largest number of parameters specified.

When performing service filtering with the "filteringCriteria" - "addressAndService", the first parameters checked will always be the "serviceKey" and "calledAddressValue".

If an "ActivateServiceFiltering" operation is passed to the SSF with the "filteringCriteria" "addressAndService" with both callingAddressValue and "locationNumber" present, the following is applicable:

– When the SSF receives a call that matches "serviceKey" and "calledAddressValue" (in the active "filteringCriteria"), it investigates whether or not the locationNumber" is present in the initial address message. If it is present and matches the active “filteringCriteria” the call is filtered. If the SSF finds that the “locationNumber” is absent, then it will check the “callingAddressValue” and perform filtering depending on that parameter.

If no errors occurred after receiving an “ActivateServiceFiltering” on the SSF, an empty Return Result is sent to the SCF. That causes no state transition in the SSME-FSM.

The following application timers are used:

– detect moment to start service filtering (start time)
– duration time for service filtering
– interval time for service filtering (for timer controlled approach)

7.4.3.2.4.2 Error handling

If the SSF detects an error with any of the defined error values then this error is reported to the SCF.

The event is recorded in the SSF and an error condition indicated.

In case a new SSME FSM should be created, the relationship is ended and all concerned resources (e.g., counters) are released. The SSME FSM remains in the state “Idle Management”.

In case there is already an existing SSME FSM, the service filtering data remain unchanged. The SSME FSM remains in the state “Non-Call Associated Treatment”.

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.3 ActivityTest procedure

7.4.3.3.1 General description

This operation is used to check for the continued existence of a relationship between the SCF and SSF. If the relationship is still in existence, then the SSF will respond. If no reply is received within a given time period, then the SCF will assume that the SSF has failed in some way and will take the appropriate action.

7.4.3.3.2 Parameters

None.

7.4.3.3.3 Invoking entity (SCF)

7.4.3.3.3.1 Normal procedure

SCF preconditions:

1. A relationship exists between the SCF and the SSF.
2. The activity test timer(Tat) expires, after which the “ActivityTest” operation is sent to the SSF.

SCF postcondition:

If a Return Result "ActivityTest" is received, the SCME resets the activity test timer and takes no further action.

7.4.3.3.3.2 Error handling

If a time out on the "ActivityTest" operation or a P-Abort is received from TCAP, this is an indication that the relationship with the SSF was somehow lost. If a time-out is received, SCF aborts the dialogue.

The SLPI that was the user of this dialogue will be informed, and the corresponding SCSM-FSM will move to the state "idle".

7.4.3.3.4 Responding entity (SSF)

7.4.3.3.4.1 Normal procedure

SSF precondition:

A relationship exists between the SCF and the SSF.

SSF postconditions:

1. The SSME-FSM stays in, or moves to the state "Non Call Associated Treatment".

2. If the Dialogue ID is active and if there is an SSF-FSM using the dialogue, the SSME sends a Return Result "ActivityTest" to the SCF. If there are no other management activities, the SSME-FSM returns to the state "Idle Management", or

   If the Dialogue ID is not active, the TCAP in the SSF will issue a P-Abort, and the SSME will in that case never receive the "ActivityTest" result and thus will not be able to reply.

7.4.3.3.4.2 Error handling

Operation related error handling is not applicable, due to class 3 operation.

7.4.3.4 AnalyzedInformation procedure

7.4.3.4.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the Analyzed_Information DP, or to report an event requested by RequestReportBCMEvent.

7.4.3.4.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. This parameter identifies the originating facility.

- bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.
- calledPartyNumber *(CalledPartyID)*:
  See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.
- servingAreaID *(Lata)*:
  See 3.1.
- serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
  See 7.4.6 for population rules for serviceAddressInformationTriggerType.
- chargeNumber *(ChargeNumber)*:
  See 3.1. See 7.4.6 for population rules for chargeNumber.
- callingPartyNumber *(CallingPartyID)*:
  See T1.113.
- callingPartyBusinessGroupID *(CallingPartyBGID)*:
  See 3.1. The SCF can use this information element to select SLPs based on the group and for authorization purposes. The network operators can specify that this information element should be used if their particular network has the information available.
- callingPartysCategory *(ChargePartyStationType)*:
  See T1.113.
- carrier *(Carrier)*:
  See 3.1.
- accessCode *(AccessCode)*:
  See 3.1.
- dialedDigitsCAI *(CollectedAddressInfo)*:
  See 3.1. See 7.4.6 for population rules for dialedDigitsCAI.
- dialedDigitsCD *(CollectedDigits)*:
  See 3.1. See 7.4.6 for population rules for dialedDigitsCD.
- featureCode *(VerticalServiceCode)*:
  See 3.1.
- travelingClassMark *(Tcm)*:
  See 3.1.
- originalCalledPartyID *(OriginalCalledPartyID)*:
  See T1.113 Original Called Number signaling information.
- redirectingPartyID *(RedirectingPartyID)*:
  This parameter (if available) is the directory number of the last redirecting party.
- redirectionInformation *(RedirectionInformation)*:
  See T1.113 Redirection Information signaling information.
- cGEncountered *(ACGEncountered)*:
  See 3.1.
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

- sap *(Sap)*:
  When the originating access is an SS7 trunk and the received IAM contains the Sap parameter, the SSF shall map the Sap parameter received in the IAM directly to the corresponding Sap parameter in the INAP message.
- sTRConnection *(STRConnection)*:
  This parameter is sent to indicate whether or not the event being reported resulted from a Send_To_Resource operation.
- aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.
- genericAddressList *(GenericAddressList)*:
  This parameter is a series of Generic Address as defined in T1.113.
- networkSpecificFacilities *(NetworkSpecificFacilities)*:
  This parameter is sent to identify the particular service being delivered to the PRI.
- forwardCallIndicator *(ForwardCallIndicator)*:
  This parameter indicates the signaling capabilities of the network access, preceding network connection, and the preferred signaling capabilities for the succeeding network connection. It also contains information for the Call Completion to Portable Number (CCPN) feature.
- genericDigitsList *(GenericDigitsList)*:
  This parameter includes a GenericDigits parameter that contains signaled Location Information for a wireline Emergency Caller or cell site and sector information for a wireless Emergency Caller (See T1.628 for Emergency Calling Service description and usage).
- callingGeodeticLocation *(CallingGeodeticLocation)*:
  This parameter contains the latitude and longitude information of an Emergency Caller (See T1.628 for Emergency Calling Service description and usage).

7.4.3.4.3 Invoking entity (SSF)

7.4.3.4.3.1 Normal procedure

SSF preconditions (TDP):

1. Call origination attempt has been initiated.
2. *Called Party Number* is available and nature of address determined.
3. Call gapping or service filtering is not in effect.
4. DP criteria are met.
5. For a TDP-R, there is no existing control relationship influencing the call segment.

SSF preconditions (EDP):

1. For an EDP-R, there is an existing control relationship and the EDP AnalyzedInformation is armed.
2. For an EDP-N, there is a monitoring or control relationship.

SSF postconditions (TDP):

1. For a TDP-R, basic call processing has been suspended at Analyzed_Information DP, and a control relationship has been established.
2. For a TDP-N, basic call processing proceeds at Select_Route PIC, and no control relationship is established.

SSF postconditions (EDP):

1. For an EDP-R, basic call processing has been suspended at Analyzed_Information DP, and the existing control relationship continues.
2. For an EDP-N, basic call processing proceeds at Select_Route PIC, and the existing non-control relationship continues unless no further EDPs are armed and no “CallInformationReport” or “ApplyChargingReport” is requested.

The SSF has sufficient information available associated with the originating call portion. This information has been analyzed and the results are as described below.

1. From a non-ISDN line or ISDN interface, the Analysis Results consists of one or more of the following:

– CalledPartyID - The number used to identify the called party in the forward direction (i.e., it is used to populate the bearer signaling protocol’s *Called Party Number* information element).

– TypeOfCall - Indicates one of: interexchange carrier, international carrier, local exchange operator, interexchange carrier operator or international carrier operator.

– Carrier (for calls requiring an interexchange carrier) - Indicates the type of carrier requested.

– CarrierIdentificationCode (for calls requiring an interexchange carrier) - Indicates the code for the specific carrier to be used.

– CarrierSelection (for calls requiring an interexchange carrier) - Indicates whether the caller dialed the selected carrier and whether the caller presubscribed to the selected carrier.

– RouteIndex - An index to a route list (if the call does not terminate on this SSF).

– Collected Information - Access Codes and Prefixes, Collected Address Information/Digits.

2. From a conventional or SS7-supported trunk interface, this consists of one or more of the following:

– ChargeNumber

– CalledPartyID

– TypeOfCall

– CarrierIdentificationCode (for interexchange carrier calls)

– Carrier (for interexchange carrier calls)

– CarrierSelection (for interexchange carrier calls)

– RouteIndex

– Collected Information - Collected Address Information, Prefixes, etc. The Collected Information for a call from an SS7-supported trunk is based on information provided in the IAM.

7.4.3.4.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.4.4 Responding entity (SCF)

7.4.3.4.4.1 Normal procedure

SCF preconditions (TDP):

None

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. The SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” are requested, or
3. The SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.4.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.5 AnalyzeRoute procedure

7.4.3.5.1 General description

This operation requests the SSF to resume call origination processing taking into account the address, routing and billing information provided in the operation parameters.

7.4.3.5.2 Parameters

- chargeNumber *(ChargeNumber)*:
  See 3.1.
- callingPartyNumber *(CallingPartyID)*:
  See T1.113.
- callingPartysCategory *(ChargePartyStationType)*:
  See T1.113.
- calledPartyNumber *(CalledPartyID)*:
  See T1.113. Either the calledPartyNumber or the destinationRoutingAddress shall be provided by the SCF in the AnalyzeRoute operation.
- destinationRoutingAddressON *(OutpulseNumber)*:
  This parameter represents the number to be outpulsed.
- travelingClassMark *(Tcm)*:
  See 3.1.
- routeListPTG *(PrimaryTrunkGroup)*:
  This parameter represents the primary route to be used.
- routeListATG *(AlternateTrunkGroup)*:
  This parameter represents the alternate route to be used.
- routeListSATG *(SecondAlternateTrunkGroup)*:
  This parameter represents the second alternate route to be used.
- carrier *(Carrier)*:
  This parameter represents the primary carrier to be used.
- carrierAC *(AlternateCarrier)*:
  This parameter represents the alternate carrier to be used.
- carrierSAC *(SecondAlternateCarrier)*:
  This parameter represents the second alternate carrier to be used.
- alertingPattern *(PassiveLegTreatment)*:
  See 3.1. This parameter only applies if the network signaling supports this parameter or if the SSF is the terminating local exchange for the subscriber.
- redirectingPartyID *(RedirectingPartyID)*:
  This parameter contains the directory number of the last redirecting party.
- primaryBillingIndicator *(PrimaryBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.
- alternateBillingIndicator *(AlternateBillingIndicator)*:
  This parameter is sent to indicate the AMA call type and service feature identifier for the alternate trunk group.
- secondAlternateBillingIndicator *(SecondAlternateBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the second alternate trunk group.
- overflowBillingIndicator *(OverflowBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the carrier that is used to route the call.
- aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
  This parameter is sent to identify an alternate billing number to which the IN service should be billed.

- aMABusinessCustomerID *(AMABusinessCustomerID)*:
  This parameter is sent to identify the business customer and the type of customer ID.
- aMALineNumber *(SEQUENCE OF AMALineNumber)*:
  This parameter may include information such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).
- aMAslpID *(AMAslpID)*:
  This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.
- aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
  This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- serviceProviderID *(ServiceProviderID)*:
  This parameter identifies the Message Storage and Retrieval System.
- serviceContext *(ServiceContext)*:
  This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).
- aMABillingFeature *(AMABillingFeature)*:
  This parameter is reserved for future use.
- aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.
- redirectionInformation *(RedirectionInformation)*:
  See T1.113 Redirection Information signaling information.
- carrierUsage *(CarrierUsage)*:
  This parameter indicates to the SSF how and when to use the carrier information contained in the Carrier, AlternateCarrier and SecondAlternateCarrier parameters.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.
- genericAddressList *(GenericAddressList)*:
  This parameter is a series of Generic Address as defined in T1.113.
- networkSpecificFacilities *(NetworkSpecificFacilities)*:
  This parameter is sent to identify the particular service being delivered to the PRI.
- callingPartyBGID *(CallingPartyBGID)*:
  This parameter is sent to provide Business Group information associated with the originator or redirecting party.
- forwardCallIndicator *(ForwardCallIndicator)*:
  This parameter indicates the signaling capabilities of the network access, preceding network connection, and the preferred signaling capabilities for the succeeding network connection. It also contains information for the CCPN feature.

- aMAServiceProviderID *(AMAServiceProviderID)*:
  This parameter is sent to provide the identifier of a service provider when the service logic resides on an SCP and the service provider is not the company whose equipment is generating the AMA.
- destinationRoutingAddress *(DestinationRoutingAddress)*:
  See T1.113. Either the calledPartyNumber or the destinationRoutingAddress shall be provided by the SCF in the AnalyzeRoute operation. DestinationRoutingAddress contains a list of CalledPartyIDs.
- destinationRoutingAddressUsage (*DestinationRoutingAddressUsage*):
- This parameter represents the events that shall be monitored by the SSF while processing the *CalledPartyIDs* from the *DestinationRoutingAddress parameter*.

7.4.3.5.3 Invoking entity (SCF)

7.4.3.5.3.1 Normal procedure

The SCF shall populate the parameters and send this operation based on Service Logic.

7.4.3.5.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.5.4 Responding entity (SSF)

7.4.3.5.4.1 Normal Procedure

This operation requests the SSF to resume call origination processing taking into account the address, routing and billing information provided in the operation parameters. When provided in the operation, trunk group parameters containing route indices and carrier parameters determine the point where call processing resumes within the Originating BCSM. The contents of the carrier parameter further determine whether or not trigger processing within the Analyzed_Information TDP will be performed. When address information alone is provided in the operation, processing resumes at the Analyze_Information PIC.

The SSF response to the receipt of the AnalyzeRoute operation when routing parameters are included in the operation is as follows:

When a trunk group is specified and used to attempt to route the call, the SSF resumes call processing at the Select_Route PIC.

When a carrier is specified and used in an attempt to route the call, call processing resumes at the beginning of the Analyze_Information PIC. In this case, trigger analysis is only performed if the selected carrier is identified with a Carrier ID of 0110.

The following operation parameter configurations are supported and shall result in the SSF call processing actions specified:

No Trunk Group Specified - No Carrier Specified

The SSF shall resume call processing at the Analyze_Information PIC.

Switch-based carrier selection shall be performed.

The switch shall use either the carrier designated by the 101XXXX Carrier Access Code (CAC) dialed by the user or the user's presubscribed carrier when a CAC code is not dialed.

The called party address provided in the *CalledPartyID* parameter shall be subjected to trigger analysis.

One or More Trunk Groups Specified - No Carrier Specified

The SSF shall resume call processing at the Select_Route PIC and proceed according to the BCSM description to attempt to complete the call over the designated route(s).

No Trunk Groups Specified - One or More Carriers Specified

The SSF shall accept an AnalyzeRoute operation with one or more *Carrier* parameters, each specifying a 4-digit Carrier Identification Code (XXXX or 0XXX).

The SSF shall resume call processing at the Analyze_Information PIC using the designated carrier(s).

If this carrier is the LEC, trigger analysis shall be performed.

If the selected carrier is an IC, trigger processing shall be determined by the network operator.

This action shall be repeated, if the selected carrier is unavailable, using the alternate carriers specified in the AnalyzeRoute operation.

One or More Trunk Groups Specified - One or More Carriers Specified

The SSF shall resume call processing at the Select_Route PIC and proceed through the list of designated trunk groups as described above.

If the call cannot be completed using the trunk groups specified, the SSF shall return call processing to the Analyze_Information PIC and attempt to complete the call via the designated carriers as described above.

The AnalyzeRoute operation shall be able to specify a maximum of three trunk groups and three carriers. The SSF shall attempt to route the call over the designated trunk groups and carriers in the following order:

Primary Trunk Group

Alternate Trunk Group

Second Alternate Trunk Group

Carrier

Alternate Carrier

Second Alternate Carrier.

Alternate route/carrier selection as indicated in the SCF Response or Conversation Package shall proceed until an idle facility is found, processing of the call encounters a screening feature that blocks the call, or a subsequent IN trigger or requested event is detected.

If the *DestinationRoutingAddress* parameter is included instead of the *CalledPartyID*, the SSF shall use each *CalledPartyID* from the *DestinationRoutingAddress* parameter one at a time with each TrunkGroup and Carrier parameter. Alternate routing among TrunkGroup and Carrier parameters would be based on route busy conditions (local or remote) as is currently done (as defined in 5.3.2.1.7). Choosing another *CalledPartyID* from the list would be based on the defined Busy, No Answer, or Network Busy events specified in the DestinationRoutingAddressUsage parameter. That is, in general, the first *CalledPartyID* in the *DestinationRoutingAddress* parameter would be tried for every SCP supplied TrunkGroup and Carrier parameter, then the second *CalledPartyID*, etc.

More specifically, alternate trunk groups and carriers would be tried for a specific *CalledPartyID* from the *DestinationRoutingAddress* parameter only if the route busy conditions (local or remote) for alternate routing occur. That is, once an idle facility was located to route the call to the DN, no attempt would be made to continue to hunt another idle facility based on the Busy or No Answer events, i.e., Busy or No Answer would not result in alternate routing.

To further illustrate, listed below is a summary of the processing that would occur:

1. Select first *CalledPartyID* from the *DestinationRoutingAddress* parameter
2. Route to selected *CalledPartyID* over 1st SCP provided route
3. Alternate route via next SCP supplied routes only if an idle facility could not be found (e.g. a route busy condition occurred)
4. Once an idle facility is found, route to the DN specified and monitor for the Busy or No Answer event(s) indicated in the *DestinationRoutingAddressUsage* parameter.
5. If the Busy or No Answer events are satisfied, select the next CalledPartyID and start the process again at step 2

Chosing a new CalledPartyID from the DestinationRoutingAddress based on the NetworkBusy event armed in the DestinationRoutingAddressUsage parameter is illustrated with the following processing summary:

1. Select the first CalledPartyID from the DestinationRoutingAddress parameter
2. Route to selected CalledPatyID over 1st SCP provided route
3. Alternate route via next SCP supplied route if an idle facility could not be found (e.g. a route busy condition occurred either on the local or remote switch)
4. If all SCP supplied routes have been exhausted and no idle facility could be found, and the "networkBusy" event was armed in the DestinationRoutingAddressUsage parameter, select the next CalledPartyID and start the process again at step 2.

The *DestinationRoutingAddressUsage* parameter contains the following information for the SSF to monitor the specified event(s) and decide when to choose the next *CalledPartyID* from the *DestinationRoutingAddress* parameter.

- NetworkBusy? (Y/N) – Select the next CalledPartyID if all carriers and trunk groups provided by the SCP are unavailable.
- User Busy? (Y/N) – Select the next CalledPartyID if the called party is busy.
- No Answer? (Y/N) – Select the next CalledPartyID if the called party is alerting and does not answer before the NoAnswer Timer expires.
- NoAnswer Timer

The *DestinationRoutingAddress* parameter and the *DestinationRoutingAddressUsage* parameter should not be used with the RequestReportBCMEvent in the same message.

Use of the *Carrier* parameter in conjunction with the *CalledPartyIDs* from the *DestinationRoutingAddress* parameter would be based on existing procedures as defined for the *CarrierUsage* parameter.

Table 12 shows examples of how the *CalledPartyID* and *Carrier* parameters (i.e., *Carrier*, *AlternateCarrier*, and *SecondAlternateCarrier*) are populated to identify intended call destinations.

Table 12 - Routing table

Type of Call	*CalledPartyID*		*Carrier*
	Digits	*Nature of Number*	
IntraLATA	*10D*	*National Number*	*0110*
InterLATA	*10D*	*National Number*	*XXXX*
International	*7-15D*	*International*	*XXXX*
IntraLATA 0+	*10D*	*National Number, Operator Requested*	*0110*
InterLATA 0+	*10D*	*National Number, Operator Requested*	*XXXX*
International 01+	*1-15D*	*International, Operator Requested*	*XXXX*
IntraLATA Operator	*0D*	*No Address Present, Operator Requested*	*0110*
InterLATA Operator	*0D*	*No Address Present, Operator Requested*	*XXXX*
International Operator	*0D*	*No Address Present, Operator Requested*	*XXXX*
IC 800	*800+7D*	*National*	*XXXX*
Database 800	*800+7D*	*National*	*0110*
900	*900+7D*	*National*	*XXXX*
700	*700+7D*	*National*	*XXXX*
700 Operator Assist	*700+7D*	*National, Operator Requested*	*XXXX*
N11	*N11*	*National*	*0110*
950-XXXX	*950-XXXX*	*950 + Call*	*N/A*
101XXXX-	*N/A*	*No Address Present, Carrier Cut Through*	*XXXX*

The SCF can send an AnalyzeRoute operation containing one or more *TrunkGroup* parameters to route the call or a physical or simulated private facility. In this case, each *TrunkGroup* parameter (if included) specifies whether the route index points to physical or simulated facilities and whether that route uses the *CalledPartyID* or *OutpulseNumber* parameter. In addition, the AnalyzeRoute operation may contain Traveling Class Mark (TCM) information to be included in the outgoing signaling.

If the *TrunkGroup* parameter indicates “normal routing number,” the *CalledPartyID* parameter shall be used.

If the *TrunkGroup* parameter indicates “outpulsing number,” the *OutpulseNumber* parameter shall be used.

If a route requires a TCM, the value in the *Tcm* parameter shall be used. If a TCM is required and the *Tcm* parameter is not supplied in the SCF Response or Conversation Package, the SSF shall derive the TCM in accordance with requirements that apply assuming no IN involvement with the call.

If the AnalyzeRoute operation contains a *Tcm* parameter and the call is routed on an SS7 supported trunk, the SSF shall include the Tcm value in an SS7 IAM Remote Operations parameter.

If the AnalyzeRoute operation contains a *Tcm* parameter and the call is routed over a PRI, the SSF shall include the Tcm value in a SETUP message Facility information element.

The trunk group (route index) returned by the SCF in the *PrimaryTrunkGroup, AlternateTrunkGroup, or SecondAlternateTrunkGroup* parameters shall be used by the SSF to identify either a single trunk group or the initial trunk group in an ordered list of trunk groups.

When the route index points to the initial trunk group in an ordered list and the SSF determines that this trunk group is unavailable, the SSF shall hunt through the list of trunk groups until an available trunk is found.

If the *ChargePartyStationType* parameter is included, that value shall be used as the MF II digits or the ISUP OLI parameter in any subsequent signaling.

When MF II digits are needed for subsequent signaling, the decimal equivalent of the *ChargePartyStationType* value shall be used (e.g., “01000000” is equivalent to II digits of “64”).

If the *ChargePartyStationType* parameter is not included, the SSF shall use the ChargePartyStationType information that the SSF had included in the previous Query Message to the SCF.

If the *ChargeNumber* parameter is included, that value shall be used as the MF ANI or the ISUP Charge Number parameter in any subsequent signaling.

If the *ChargeNumber* parameter is not included, the SSF shall use the ChargeNumber information that the SSF had included in the previous Query Message to the SCF.

If the *PassiveLegTreatment* parameter is included, the designated distinctive alerting pattern shall be applied as follows to the called non-ISDN line if that line is served by the SSF:

> alertingOnPattern0Normal - the SSF shall apply normal alerting, i.e., 2 seconds of ringing followed by 4 seconds of silence.
>
> alertingOnPattern1DistinctiveIntergroup - the SSF shall apply 800 ms of ringing, 400 ms of silence, 800 ms of ringing, followed by 4 seconds of silence.
>
> alertingOnPattern2DistinctiveIntergroup - the SSF shall apply 400 ms of ringing, 200 ms of silence, 400 ms of ringing, 200 ms of silence, 800 ms of ringing, followed by 4 seconds of silence.
>
> any other value - the SSF shall apply normal alerting, i.e., 2 seconds of ringing followed by 4 seconds of silence.

Note that for a called BRI, apply means to include the appropriate signal value in the ISDN SETUP message for the call.

If the *CallingPartyID* parameter is included in the AnalyzeRoute operation, that value shall be used by all features subsequently acting on that call including ISDN features, CLASS features, ISUP signaling, and other IN features.

When the SSF receives an AnalyzeRoute operation that contains the *RedirectingPartyID* parameter, the SSF shall map the received *RedirectingPartyID* parameter to:

> the Redirecting Number information element in the SETUP message if the SSF routes the call to an ISDN BRI or PRI. If the call has been forwarded at least once before, the SSF shall send the specified Redirecting Number in the Redirecting Number information element containing the last forwarding directory number
>
> the *Redirecting Number* parameter in the IAM if the SSF already has a value for the Original Called Number AND if the SSF routes the call to an SS7 trunk (for ISUP signaling)
>
> the *Original Called Number* parameter in the IAM if the SSF does not already have a value for the Original Called Number AND if the SSF routes the call to an SS7 trunk (for ISUP signaling).

7.4.3.5.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.6 ApplyCharging procedure

7.4.3.6.1 General description

This operation is used for interacting from the SCF with the SSF charging mechanisms. The "ApplyChargingReport" operation provides the feedback from the SSF to the SCF.

As several connection configurations may be established during a call, a possibility exists for the "ApplyCharging" to be invoked at the beginning of each connection configuration, for each party.

7.4.3.6.2 Parameters

– aChBillingChargingCharacteristics *(AChBillingChargingCharacteristics)*:
  This parameter specifies the charging related information to be provided by the SSF and the conditions on which this information has to be reported back to the SCF via the "ApplyChargingReport" operation. Its content is network operator-specific.

– partyToCharge *(PartyToCharge)*:
  This parameter indicates the party in the call to which the "ApplyCharging" operation should be applied. If this parameter is not present, then the "ApplyCharging" operation is applied to the A-party.

– extensions *(SEQUENCE OF ExtensionField)*:
  This parameter indicates an extension of an argument data type. Its content is network-specific.

7.4.3.6.3 Invoking entity (SCF)

7.4.3.6.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. The SLPI has determined that an "ApplyCharging" operation has to be sent.

SCF postconditions:

1. No FSM state transition.
2. The SLPI is expecting "ApplyChargingReport" operations from the SSF.

The SCSM FSM is in state "Preparing SSF Instructions" or is in state "Queuing FSM". This operation is invoked by the SCF if an SLPI results in the request of interacting with the charging mechanisms within the SSF to get back information about the charging.

7.4.3.6.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.6.4 Responding entity (SSF)

7.4.3.6.4.1 Normal procedure

SSF preconditions:

The SSF-FSM is in one of the following states:
"Waiting for Instructions" (state c),
"Waiting for End of User Interaction" (state d),
"Waiting for End of Temporary Connection" (state e),
"Monitoring" (state f),
or the assisting/hand-off SSF-FSM is in state:
"Waiting for Instructions" (state b)

SSF postconditions:

No FSM state transition

On receipt of this operation, the SSF sets the charging data using the information elements included in the operation and acts accordingly. In addition, the SSF will start the monitoring of the end of the connection configuration and other charging events, if requested.

7.4.3.6.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.7 ApplyChargingReport procedure

7.4.3.7.1 General description

This operation is used by the SSF to report charging related information to the SCF as requested by the SCF using the "ApplyCharging" operation.

During a connection configuration the `ApplyChargingReport" operation may be invoked on multiple occasions. For each call party and each connection configuration, the "ApplyChargingReport" operation may be used several times. Note that at least one "ApplyChargingReport" operation is to be sent at the end of the connection configuration charging process.

7.4.3.7.2 Parameters

– callResult:
  This parameter provides the SCF with the charging related information previously requested using the "ApplyCharging" operation. The "callResult" will include the "partyToCharge" parameter as received in the related "ApplyCharging" operation to correlate the result to the request. The remaining content of "callResult" is network operator-specific. Examples of these contents may be: bulk counter values, costs, tariff change, and time of change, time stamps, durations, etc.

7.4.3.7.3 Invoking entity (SSF)

7.4.3.7.3.1 Normal procedure

SSF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. A charging event has been detected that was requested by the SCF via an “ApplyCharging” operation.

SSF postconditions:

If the connection configuration does not change then no FSM state transition shall occur.

If the connection configuration changes then the FSM shall move to:

– “Idle” state if there is no other EDP armed and no report requests are pending, or otherwise;

– shall remain in the same state.

This operation is invoked if a charging event has been detected that was requested by the SCF. The “ApplyChargingReport” operation only deals with charging events within the SSF itself. Examples of charging events may be: threshold value reached, timer expiration, tariff change, end of connection configuration, etc.

7.4.3.7.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.7.4 Responding entity (SCF)

7.4.3.7.4.1 Normal procedure

SCF preconditions:

An “ApplyCharging” operation has been sent at the request of an SLPI and the SLPI is expecting an “ApplyChargingReport” from the SSF.

SCF postconditions:

No FSM state transition if further reports, including reports of requested events and “CallInformationReport”, are expected, or Transition to the state “Idle” if the report is the last one and reports of requested events or “CallInformationReport” is expected.

On receipt of this operation the SLPI that is expecting this operation will continue.

7.4.3.7.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.8 AuthorizeTermination procedure

7.4.3.8.1 General description

The AuthorizeTermination operation requests the SSF to continue processing at the Authorize_Termination_Attempt PIC (i.e., to verify the authority to route the call to the directory number/call type for which the trigger was encountered). After call processing has resumed, the call may subsequently encounter terminating switch-based features.

7.4.3.8.2 Parameters

– callingPartyNumber *(CallingPartyID)*:
 See T1.113.

– travelingClassMark *(Tcm)*:
 See 3.1.

– alertingPattern *(ControllingLegTreatment)*:
 See 3.1. This parameter only applies if the network signaling supports it or if the SSF is the terminating local exchange for the subscriber.

– displayInformationDT *(DisplayText)*:
 This parameter indicates display text to be sent to the end-user.

– primaryBillingIndicator *(PrimaryBillingIndicator)*:
 This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.

– aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
 This parameter is sent to identify an alternate billing number to which the IN service should be billed.

– aMABusinessCustomerID *(AMABusinessCustomerID)*:
 This parameter is sent to identify the business customer and the type of customer ID.

– aMALineNumber *(SEQUENCE OF AMALineNumber)*:
 This parameter may include information such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).

– aMAslpID *(AMAslpID)*:
 This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.

– aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
 This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).

– amp1 *(Amp1)*:
 This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
 This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– serviceProviderID *(ServiceProviderID)*:
 This parameter identifies the Message Storage and Retrieval System.

– serviceContext *(ServiceContext)*:
  This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).

– aMABillingFeature *(AMABillingFeature)*:
  This parameter is reserved for future use.

– aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

– aMAServiceProviderID *(AMAServiceProviderID)*:
  This parameter is sent to provide the identifier of a service provider when the service logic resides on an SCP and the service provider is not the company whose equipment is generating the AMA.

7.4.3.8.3 Invoking entity (SCF)

7.4.3.8.3.1 Normal procedure

The SCF shall populate the parameters and invoke these procedures based on Service Logic.

7.4.3.8.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.8.4 Responding entity (SSF)

7.4.3.8.4.1 Normal procedure

The SSF shall continue processing at the Authorize_Termination_Attempt PIC using the following information.

CallingPartyID:

Within the AuthorizeTerminationArg, if the *CallingPartyID* parameter is included, that value shall be used as the ISUP and/or ISDN Calling Party Number in any subsequent signaling or as the Calling Party Number used by any CLASS feature.

Tcm:

If the *Tcm* parameter is included, that value shall be used as the TCM (if the call is routed over a trunk that requires a TCM). If a TCM is required and the *Tcm* parameter is not supplied in the SCF Response Package, the SSF shall derive the TCM in accordance with requirements that apply assuming no IN involvement with the call.

ControllingLegTreatment:

If the ControllingLegTreatment parameter is included, the designated distinctive alerting pattern shall be applied to the called party if the called party is served by the SSF as follows:

alertingOnPattern0Normal - the SSF shall apply normal alerting, i.e., 2 seconds of ringing followed by 4 seconds of silence.

alertingOnPattern1DistinctiveIntergroup - the SSF shall apply 800 ms of ringing, 400 ms of silence, 800 ms of ringing, followed by 4 seconds of silence.

alertingOnPattern2DistinctiveIntergroup - the SSF shall apply 400 ms of ringing, 200 ms of silence, 400 ms of ringing, 200 ms of silence, 800 ms of ringing, followed by 4 seconds of silence.

any other value - the SSF shall apply normal alerting, i.e., 2 seconds of ringing followed by 4 seconds of silence.

DisplayText:

The *DisplayText* parameter, if included, contains information for display on the called party's CPE. The *DisplayText* parameter can provide one or more fields of display information (e.g., called address information would be provided in one field) to be sent to the called party; the type of information contained in each field is identified by its tag. These tags are a subset of the tags found in the ISDN display text information element.

The SSF shall expect each field of display information to consist of 1 to 20 characters (including spaces). A field identified by the continuation tag shall be associated with the field preceding it within the *DisplayText* parameter. A field of display information exceeding 20 characters (including spaces) shall be truncated to 20 characters.

When the *DisplayText* parameter is present in the AuthorizeTermination operation, the SSF sends display information to the subscriber's CPE in one of the following ways:

Call termination to an analog line:

If the call is terminating to an analog line, the SSF shall extract the callingPartyName, callingAddress, the reason, and dateTimeOfDay elements from the *DisplayText* parameter and map them to Multiple Data Message parameters. The corresponding Multiple Data Message parameters are Name, Calling Line ID, Reason for Absence, and Time parameters. The SSF shall send these parameters to the CPE using the Multiple Data Message Format.

Call termination to an ISDN Basic Rate Interface (BRI):

If the call is terminating to an ISDN BRI, the SSF shall map the *DisplayText* parameter into the display text information element and send the information element to the CPE in a SETUP or INFOrmation message.

Call termination to an ISDN Primary Rate Interface (PRI):

If the call is terminating to an ISDN PRI, the SSF shall map the following fields of the *DisplayText* parameter into the facility information element and send the information element to the CPE in a SETUP or FACILITY message:
(1) callingPartyName
(2) originalCalledName
(3) redirectingName
(4) redirectingReason.

After sending SCF-provided display text, the SSF shall continue to process the call as if the display had been generated by the switch itself. That is, switch-based display information available at a later point in call (e.g., when the call is connected) shall still be sent. In addition, the SSF shall clear the IN display text, via a display clearing request when the call is taken down.

7.4.3.8.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.9 CallInfoFromResource procedure

7.4.3.9.1 General description

This operation is sent by the SSF to the SCF in the SSF Relay case or from the SRF to the SCF in the direct SCF-SRF case to convey the results of an interaction between the user and the SRF that has occurred during an established SendToResource connection.

7.4.3.9.2 Parameters

- iPReturnBlock *(IPReturnBlock)*:
  This parameter contains the results of the user interaction with the SRF.
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.9.3 Invoking entity (SSF/SRF)

7.4.3.9.3.1 Normal procedure

– SSF Relay

SSF preconditions:

1. a control relationship exists between the SCF and the SSF
2. the SSF has established a SendToResource connection with the SRF
3. the SRF has performed the requested operation but intends to retain the established connection
4. the SSF has received a FACILITY message from the SRF either containing information that must be sent to the SCF/Adjunct or requesting additional data from the SCF/Adjunct

SSF postcondition:

The SSF shall start a timer and wait for a response to be returned from the SCF/Adjunct

The SSF shall formulate the CallInfoFromResource message and send this operation based upon the information received in the FACILITY message from the SRF. The FACILITY message must contain a Facility Information Element (FIE) with a ReturnResult component. The FACILITY message contains the call reference value established by the SETUP message for the STR connection. The SSF shall also include the optional parameter iPReturnBlock if this parameter was included in the FIE.

If the timer expires before the SSF receives a response from the SCF/Adjunct, the SSF shall initiate the release procedure to the SRF.

The SSF shall not accept the FACILITY message from the SRF if there is already an outstanding CallInfoFromResource that has not yet received a CallInfoFromResourceRR response from the SCF.

If the SSF cannot accept the FACILITY message containing the FIE with the ReturnResult component then the SSF shall clear the SendToResource connection towards the SRF, and will send a ResourceClear message to the SCF/Adjunct.

– Direct SCF-SRF

SRF preconditions:

1. a direct control relationship exists between the SCF and the SRF
2. the SRF has received an InstructionsToSRF operation from the SCF
3. the SRF has performed the requested operation
4. the SRF intends either passing the information that must be sent to the SCF or requesting additional data from the SCF

SRF postcondition:

The SRF shall start a timer and wait for a response to be returned from the SCF.

The SRF shall formulate the CallInfoFromResource message and send this operation to the SCF.

7.4.3.9.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.9.4 Responding entity (SCF)

7.4.3.9.4.1 Normal procedure

– SSF Relay

SCF preconditions:

1. a control relationship exists between the SCF and the SSF
2. the SCF is expecting either a CallInfoFromResource or a ResourceClear from the SSF

SCF postcondition:

The SLPI shall be executed to formulate the CallInfoFromResourceRR response to the SSF

The SCF shall not have an existing CallInfoFromResource for the STR-connection already in progress to which it has not yet responded.

– Direct SCF-SRF

SCF preconditions:

1. a control relationship exists between the SCF and the SRF

2. the SCF is expecting a CallInfoFromResource from the SRF

SCF postcondition:

The SLPI shall be executed to formulate the CallInfoFromResourceRR response to the SRF

The SCF shall not have an existing CallInfoFromResource for the STR-connection already in progress to which it has not yet responded.

7.4.3.9.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.10 CallInfoFromResourceRR procedure

7.4.3.10.1 General description

CallInfoFromResourceRR is sent by the SCF to the SSF/SRF as a response to a CallInfoFromResource operation. The SSF will convey the information to the SRF in the SSF Relay case. CallInfoFromResourceRR corresponds to the RESULT portion of the ASN.1 definition of the CallInfoFromResource operation.

7.4.3.10.2 Parameters

– resourceType *(ResourceType)*:
The r*esourceType* parameter identifies the specific resource type (or capability) that is to be provided to the user.

– strParameterBlock *(StrParameterBlock)*:
The s*trParameterBlock* parameter is sent to provide the information needed for the capability specified by the r*esourceType* parameter. It can take one of a number of different encodings depending on the capability that is to be used. Each resource type value implies its own parameter block and its own encoding.

– amp1 *(Amp1)*:
The a*mp1* parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– serviceProviderID *(ServiceProviderID)*:
This parameter identifies the Message Storage and Retrieval System.

– serviceContext *(ServiceContext)*:
This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– disconnectFromIPForbidden *(DisconnectFromIPForbidden)*
  In the direct SCF to SRF information transfer case, this parameter is used to allow the SRF initiated disconnect on the bearer connection if it is set to FALSE.

7.4.3.10.3 Invoking entity (SCF)

7.4.3.10.3.1 Normal procedure

– SSF Relay

SCF preconditions:

1. a control relationship exists between the SCF and SSF
2. the SCF has received a CallInfoFromResource from the SSF

SCF postcondition:

The SLPI may be further executed

The SCF shall populate the parameters and send this operation based on Service Logic.

– Direct SCF-SRF

SCF preconditions:

1. a control relationship exists between the SCF and the SRF
2. the SCF has received a CallInfoFromResource from the SRF

SCF postcondition:

The SLPI may be further executed

The SCF shall populate the parameters and send this operation based on Service Logic.

7.4.3.10.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.10.4 Responding entity (SSF/SRF)

7.4.3.10.4.1 Normal procedure

– SSF Relay

SSF preconditions:

1. a control relationship exists between the SCF and the SSF
2. the SSF has sent a CallInfoFromResource to the SCF and started a timer
3. the SSF has established a SendToResource connection with the SRF

SSF postconditions:

1. the SSF shall retire the timer
2. the SSF shall formulate the appropriate message to be sent to the SRF

Upon receiving a CallInfoFromResourceRR message in response to the CallInfoFromResource message, the SSF will form and send a FACILITY message with an Invoke component to the SRF. The SRF will perform the next request in the Invoke component and return the results in either a FACILITY message or a DISConnect message.

The SSF shall also include the parameters in the Facility Information Element (FIE) if they are present in the ReturnResult component of the CallInfoFromResourceRR message.

Upon receiving a CallInfoFromResourceRR message containing the ReturnResult component with no parameters in response to the CallInfoFromResource message, the SSF will initiate STR-connection clearing by sending a DISConnect message to the SRF.

The SSF shall then formulate and send a ResourceClear message to the SCF/Adjunct. The *clearCause* parameter shall contain a value of "normal."

– Direct SCF-SRF

SRF preconditions:

1. a direct control relationship exists between the SCF and the SRF
2. the SRF has sent a CallInfoFromResource to the SCF and started a timer

SRF postconditions:

the SRF shall retire the timer

Upon receiving a CallInfoFromResourceRR message with additional instructions and the DisconnectFromIPForbidden parameter is set to TRUE (this is the default value), the SRF will perform the next request and return the results in another CallInfoFromResource message to the SCF.

If the CallInfoFromResourceRR message contains no instructions in response to the CallInfoFromResource message and the DisconnectFromIPForbidden parameter is set to FALSE, the SRF will initiate STR-connection clearing by sending a DISConnect message to the SSF.

The SSF shall then formulate and send a ResourceClear message to the SCF. The *clearCause* parameter shall contain a value of "normal."

7.4.3.10.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.11 CallInformationReport procedure

7.4.3.11.1 General description

This operation is used to send specific call information for a single call to the SCF as requested by the SCF in previous "CallInformationRequest" operation. The report is sent at the end of a call that is indicated by one of the events specified below.

7.4.3.11.2 Parameters

- requestedInformationList (RequestedInformationList):
  According to the requested information the SSF sends the appropriate types and values to the SCF.
- correlationID (CorrelationID):
  Used for correlation with a previous operation.
- extensions (SEQUENCE OF ExtensionField):
  This parameter indicates an extension of an argument data type. Its content is network-specific.

7.4.3.11.3 Invoking entity (SSF)

7.4.3.11.3.1 Normal procedure

SSF preconditions:

1. At least one party disconnects from a call or call setup is not completed.
2. Requested call information has been collected.
3. “CallInformationReport” is pending due to a previously received “CallInformationRequest” operation.
4. A control relationship exists between the SCF and the SSF.

SSF postcondition:

The SSF FSM shall move to the “Idle” state in the case where no other report requests are pending and no EDPs are armed otherwise the SSF FSM shall remain in the same state.

If the SSF FSM executes a state transition caused by one of the following events:

- A party release
- A party abandon
- B party release
- B party busy
- SSF no answer timer expiration
- route select failure indicated by the network
- release call initiated by the SCF,

and “CallInformationRequest” is pending then one “CallInformationReport” operation is sent to the SCF containing all information requested.

If a “CallInformationReport” has been sent to the SCF then no “CallInformationReport” is pending, i.e., a further “CallInformationReport”, for example in the case of follow-on, has to be explicitly requested by the SCF.

If an event causing the “CallInformationReport” is also detected by an armed EDP-R then immediately after “CallInformationReport” the corresponding requested event report has to be sent.

If an event causing the “CallInformationReport” is also detected by an armed EDP-N then immediately before “CallInformationReport” the corresponding requested event report has to be sent.

7.4.3.11.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.11.4 Responding entity (SCF)

7.4.3.11.4.1 Normal procedure

SCF preconditions:

1. An SLPI is expecting “CallInformationReport”.
2. A control relationship exists between the SCF and the SSF.

SCF postcondition:

The SLPI may be further executed.

In any state (except “Idle”) the SCSM may receive “CallInformationReport” from the SSF, when the “CallInformationReport” is outstanding.

If “CallInformationReport” is outstanding and the service logic program indicates that the processing has been completed, the SCSM remains in the same state until it receives the “CallInformationReport” operation.

When the SCF receives the “CallInformationReport” operation and the service logic processing has been completed, then the SCSM moves to the “Idle” state.

When the SCF receives the “CallInformationReport” operation and the service logic processing has not been completed yet, then the SCSM remains in the same state (requested event report or ApplyChargingReport pending).

7.4.3.11.4.2 Error handling

If requested information is not available, a “CallInformationReport” will be sent, indicating the requested information type, but with “RequestedInformationValue” filled in with an appropriate default value as specified by the network operator.

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.12 CallInformationRequest procedure

7.4.3.12.1 General description

This operation is used to request the SSF to record specific information about a single call and report it to the SCF using the “CallInformationReport” operation.

7.4.3.12.2 Parameters

- requestedInformationTypeList *(RequestedInformationTypeList)*:
  This parameter specifies a list of specific items of information that is requested. The list may contain:
  - callAttemptElapsedTime:
    This parameter indicates the duration between the end of INAP processing of operations initiating call setup (“AnalyzeInformation”, “CollectInformation”, “Continue” and “SelectRoute”) and the received answer indication from the called party side.
  - In case of unsuccessful call setup the network event indicating the unsuccessful call setup stops the measurement of “callAttemptElapsedTime”.
  - callStopTime:
    This parameter indicates the time stamp when the connection is released.
  - callConnectedElapsedTime:
    This parameter indicates the duration between the received answer indication from the called party side and the release of the connection.
  - calledAddress:
    This parameter indicates the incoming called party address that was received by the SSF (i.e., before translation by the SCF) and is as available on the user-to-network interface or network-to-network interface and interpreted as per the numbering plan.
  - releaseCause:
    See T1.113.

Any set of these values can be requested.

- correlationID *(CorrelationID)*:
  Used for correlation with a previous operation.
- extensions *(SEQUENCE OF ExtensionField)*:
  This parameter indicates an extension of an argument data type. Its content is network-specific.

7.4.3.12.3 Invoking entity (SCF)

7.4.3.12.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. The SLPI has determined that a “CallInformationRequest” operation has to be sent by the SCF.

SCF postcondition:

The SLPI is expecting a “CallInformationReport” from SSF.

When the service logic program requests call information, the SCF sends the “CallInformationRequest” operation to the SSF to request the SSF to provide call related information.

The “CallInformationRequest” operation specifies the information items to be provided by the SSF.

7.4.3.12.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.12.4 Responding entity (SSF)

7.4.3.12.4.1 Normal procedure

SSF preconditions:

1. Call origination attempt has been initiated.
2. A control relationship exists between SSF and SCF.

SSF postconditions:

1. Requested call information is retained by the SSF.
2. The SSF is waiting for further instructions.

The SSF may receive the “CallInformationRequest” operation within an existing call associated (CA) dialogue only.

The “CallInformationRequest” operation is accepted by the SSF Finite State Machine (SSF-FSM) only in the state “Waiting for Instructions”. The operation does not lead to any transition to another state.

The SSF allocates a record and stores the requested information if already available and prepares the recording of information items that will become available later, such as “callStopTimeValue”.

7.4.3.12.4.2 Error handling

In any other than the “Waiting for Instruction” state the “CallInformationRequest” operation will be handled as an error with the error code “UnexpectedComponentSequence”.

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.13 CancelResourceEvent procedure

7.4.3.13.1 General description

This operation is sent by the SCF to the SSF to cancel a previous request to route the call to a resource.

7.4.3.13.2 Parameters

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– serviceProviderID *(ServiceProviderID)*:
  This parameter identifies the Message Storage and Retrieval System.

– serviceContext *(ServiceContext)*:
  This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.13.3 Invoking entity (SCF)

7.4.3.13.3.1 Normal procedure

The SCF shall populate the parameters and send this operation based on Service Logic. The SCF/Adjunct has the capability to request an SSF to terminate the connection between a user and an IP. A message sent from the SCF/Adjunct (i.e., CancelResourceEvent) requests the SSF to disconnect the IP from the connection.

7.4.3.13.4 Error Handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.13.5 Responding entity (SSF)

7.4.3.13.5.1 Normal procedure

The SSF shall inform the SRF to cancel the identified resource/announcement and shall send the SCF a Resource_Clear with the *ClearCause* indicating “resourceCancelled.”

The SSF sends a “cancelIPResource” operation in a FACILITY message to the IP when it receives a CancelResourceEvent message from the SCF/Adjunct, or the calling user hangs up when the STR-connection is active. The SSF expects the IP to send a DISConnect message when the IP receives the “cancelIPResource” operation in a FACILITY message.

The SSF treats CancelResourceEvent message as a request to terminate the STR-connection to the IP. The SSF also initiates clearing of the STR-connection if timer TSTRC expires or the calling user abandons the call while the STR-connection is established.
If the SSF receives a CancelResourceEvent message after processing the SendToResource message, but before the SSF has completed the STR-connection to the IP (i.e., the SETUP message has been sent, but the STR-connection is not active because the CONNect ACKnowledge message has not been sent to the IP), the SSF shall do the following:

A. ISDN PRI Class II-to-ISDN PRI Calls
   In the case of ISDN-to-ISDN calls, abandoned calls occur when the Stored Program Controlled Switching System (SPCS) receives a call clearing message from the calling Class II equipment any time before it has sent a CONNect ACKnowledge message to the called Class II equipment. The SPCS receives a call clearing message (i.e., DISConnect, RELease, or RELease COMplete) from the calling Class II equipment when the calling user has abandoned the call attempt. If the SPCS is ready to clear the call it shall complete call clearing at the calling interface.

When the calling ISDN user abandons the call, the SPCS shall send a DISConnect message to the called Class II equipment. The DISConnect message shall contain cause #16, “normal clearing (location: user)” or, for a packet-mode call.

For the packet mode, if the virtual call was the last virtual call on an on-demand, B-channel access connection and the calling party subscribes to conditional or unconditional notification, timer T320 shall be initiated. The SPCS should cancel timers T303, T-delay, T310, T-PROG, and T301 (if running) and shall continue call clearing at the called interface.

B. ISDN PRI Class II-to-Non-ISDN Calls
In the case of ISDN-to-non-ISDN calls, abandoned calls occur when the calling Class II equipment sends a call clearing message to the SPCS before an off-hook signal or answer supervision is monitored from the called non-ISDN line or trunk. The SPCS receives a call clearing message (DISConnect, RELease, or RELease COMplete) from the calling Class II equipment when the calling ISDN user has abandoned the call attempt. On receipt of the clearing message, if the SPCS is ready to clear the call (e.g., there are no feature operation constraints to clearing the call as in as in LSSGR FSD 15-01-0000, Basic 911 Emergency Service), it shall complete clearing of the call at the calling interface. If the SPCS is providing ringing to the called non-ISDN line, ringing shall cease and the non-ISDN line returned to idle. If the call has been routed over a non-ISDN trunk, the SPCS shall release the non-ISDN trunk connection.

C. Non-ISDN-to-ISDN PRI Class II Calls
In the case of non-ISDN lines or trunks terminating to an ISDN interface, abandoned calls occur when the SPCS monitors an on-hook signal from the non-ISDN line or release of the non-ISDN trunk before it has sent a CONNect ACKnowledge message to the called Class II equipment. On detecting an on-hook or release signal from the originating non-ISDN line or trunk, the SPCS shall send a DISConnect message to the called Class II equipment. If the call originated from a non-ISDN line, the DISConnect message shall contain cause #16 “normal clearing (location: user).” If the call originated from a non-ISDN trunk, the DISConnect message shall contain cause #127, “interworking, unspecified (location: network beyond the interworking point).” For the packet mode, if the virtual call was the last virtual call on an on-demand, B-channel access connection and the calling party subscribes to conditional or unconditional notification, timer T320 shall be initiated.

After receiving a SendToResource message in a Conversation Package from the SCF/Adjunct, but before sending a Resource_Clear to the SCF/Adjunct, the SSF shall accept a CancelResourceEvent message from the SCF/Adjunct.

The SSF shall also send a Resource_Clear message in a Conversation Package to the SCF/Adjunct.

– • If there is an outstanding FACILITY (from the SSF to the IP) when the SSF receives the CancelResourceEvent, and the DISConnect message from the IP includes an FIE with a Return Result component with the IPReturnBlock parameter, then the SSF shall include the IPReturnBlock parameter in the Resource_Clear message. The Resource_Clear message shall include a *ClearCause* parameter with value “resourceCancelled.”

– • If there is NO outstanding FACILITY (from the SSF to the IP) when the SSF receives the CancelResourceEvent, and the SSF receives a DISConnect message from the IP without an FIE, then the Resource_Clear message shall include a *ClearCause* parameter with value “resourceCancelled.”

– • If there is NO outstanding FACILITY (from the SSF to the IP) when the SSF receives the CancelResourceEvent, and the SSF receives a DISConnect message from the IP with an FIE, then the SSF shall detect that this is a protocol error. The Resource_Clear message shall include a *ClearCause* parameter with value “protocolError.” The Resource_Clear message shall not include the IPReturnBlock parameter.

The SSF receives CancelResourceEvent message from the SCF/Adjunct when the STR-connection is active between the SSF and the IP. The SSF requests the IP to clear a call by sending a FACILITY message with cancelIPResource operation when it receives a CancelResourceEvent message from the SCF/Adjunct or the calling user hangs up. The IP sends a DISConnect message with a Return Result component.

The local SSF initiates the call clearing message sequence when it receives a CancelResourceEvent message or when the user abandons the call. The message flow for the SCF/Adjunct initiating the disconnect of a remote IP is the same as the message flow for the SCF/Adjunct initiating the disconnect of a local IP. If the SCF/Adjunct determines that the STR-connection between the user and the remote IP should be terminated, it sends a CancelResourceEvent message to the local SSF. The CancelResourceEvent message requests the local SSF to terminate the STR-connection to the remote IP and to return any collected information to the SCF/Adjunct in a Resource_Clear message.

If the local SSF receives a CancelResourceEvent message from the SCF/Adjunct, it shall cancel the timer T1, if running, and initiate the release of the circuit-switched connection to the remote SSF by forming and sending a FAC message with a RO parameter for the outgoing circuit. The RO parameter shall contain an Invoke component, which shall contain an Invoke ID Information Element (with an Invoke ID not currently being used) and an Operation Code Information Element (with a “global” Operation Code Tag and a “cancelIPResource” Operation Code).

If the local SSF receives a REL message, and if the clearing of the STR-connection is in response to the SCF/Adjunct sending the local SSF a CancelResourceEvent message, the local SSF Resource_Clear message that the local SSF forms and sends to the SCF/Adjunct shall be in a Conversation Package and have a “resourceCancelled” *ClearCause*.

If the local SSF receives a REL message that contains a RO parameter with a Return Error component and an “ipTimeout” Error Code, and if the local SSF has not previously received a CancelResourceEvent message from the SCF/Adjunct for this STR-connection and has not previously detected that the user abandoned the call, the local SSF shall take one of the following actions:

– If timer T1 is not running, the local SSF shall form the Resource_Clear message and send it to the SCF/Adjunct in a Conversation Package with an “ipTimeout” *ClearCause*.
– If timer T1 is running, the local SSF shall await the receipt of a response message from the SCF/Adjunct or the expiration of the timer T1, and should take one of the following actions:
  – If the local SSF receives a response message from the SCF/Adjunct, it shall cancel timer T1 and discard the response message. The Resource_Clear message that the local SSF forms and sends to the SCF/Adjunct shall be in a Conversation Package and have an “ipTimeout” *ClearCause*.
  – If timer T1 expires before the local SSF receives a response message from the SCF/Adjunct, the local SSF shall provide treatment for timer T1 expiring.

7.4.3.13.5.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7

7.4.3.14 Close procedure

7.4.3.14.1 General description

This operation is sent by the SSF to close a TCAP transaction that would otherwise be needlessly left open.

7.4.3.14.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
See 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– closeCause *(CloseCause)*:
This parameter indicates the specific reason why a Close message is sent to end a TCAP transaction. (Values include callTerminated, eDPsCompleted, unexpectedCommunication, and calledPartyAnswered).

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

7.4.3.14.3 Invoking entity (SSF)

7.4.3.14.3.1 Normal procedure

SSF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. This relationship is no longer needed, and should be closed.

SSF postcondition:

Control relationship ended.

7.4.3.14.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.14.4 Responding entity (SCF)

7.4.3.14.4.1 Normal procedure

SCF preconditions:

A control relationship exists between the SCF and the SSF.

SCF postcondition:

Control relationship ended.

7.4.3.14.4.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.15 CollectedInformation procedure

7.4.3.15.1 General description

This operation is sent by the SSF after detection of a TDP-R, TDP-N, EDP-R, or EDP-N in the originating BCSM, to indicate availability of complete initial package/dialing string from the originating party.

7.4.3.15.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– chargeNumber *(ChargeNumber)*:
  See 3.1. See 7.4.6 for population rules for chargeNumber.

– servingAreaID *(Lata)*:
  See 3.1.

– carrier *(Carrier)*:
  See 3.1.

– serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
  See 3.1. The events applicable to the Collected_Information DP are Off_Hook_Delay, Channel_Setup_PRI, and Shared_Interoffice_Trunk.

– callingPartyNumber *(CallingPartyID)*:
  See T1.113 Calling Party Number signaling information. Refer to 7.4.6 for population rules for the callingPartyNumber parameter.

– callingPartysCategory *(ChargePartyStationType)*:
  See T1.113 Calling Party Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– accessCode *(AccessCode)*:
  See 3.1.

– dialedDigitsCAI *(CollectedAddressInfo)*:
  See 3.1. See 7.4.6 for population rules for dialedDigitsCAI.

– dialedDigitsCD *(CollectedDigits)*:
  See 3.1. See 7.4.6 for population rules for dialedDigitsCD.

– featureCode *(VerticalServiceCode)*:
  See 3.1.

– travelingClassMark *(Tcm)*:
  See 3.1. Refer to 7.4.6 for population rules for the travelingClassMark parameter.

– originalCalledPartyID *(OriginalCalledPartyID)*:
See T1.113 Original Called Number signaling information. Refer to 7.4.6 for population rules for the originalCalledPartyID parameter.

– redirectingPartyID *(RedirectingPartyID)*:
This parameter contains the directory number of the last redirecting party.

– redirectionInformation *(RedirectionInformation)*:
See T1.113 Redirection Information signaling information.

– cGEncountered *(ACGEncountered)*:
See 3.1.

– amp1 (Amp1):
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– sap *(Sap)*:
When the originating access is an SS7 trunk and the received IAM contains the Sap parameter, the SSF shall map the Sap parameter received in the IAM directly to the corresponding sap parameter in the INAP message.

– genericAddressList *(GenericAddressList)*:
This parameter is a series of Generic Address as defined in T1.113.

– aMASequenceNumber *(AMASequenceNumber)*:
This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– genericDigitsList *(GenericDigitsList):*
This parameter includes a GenericDigits parameter that contains signaled Location Information for a wireline Emergency Caller or cell site and sector information for a wireless Emergency Caller (See T1.628 for Emergency Calling Service description and usage).

– callingGeodeticLocation *(CallingGeodeticLocation):*
This parameter contains the latitude and longitude information of an Emergency Caller (See T1.628 for Emergency Calling Service description and usage).

7.4.3.15.3 Invoking entity (SSF)

7.4.3.15.3.1 Normal procedure

SSF preconditions for TDP-R and TDP-N:

1. An event resulting in trigger criteria being met has occurred at the Collected_Information DP (e.g., Off_Hook_Delay, Channel_Setup_PRI, Shared_Interoffice_Trunk).
2. Call gapping and SS7 overload are not in effect for the call, and the call is not to be filtered.

SSF preconditions for EDP-R and EDP-N:

1. A control relationship has been established and the SCF requests Collected_Information as EDP-R or EDP-N.
2. Call processing reaches the Collected_Information DP.

SSF postcondition for TDP-R:

A control relationship has been established and the SSF waits for instructions from the SCF.

SSF post-condition for EDP-R:

The control relationship continues and the SSF waits for instructions from the SCF.

SSF post-condition for TDP-N and EDP-N:

Call processing continues at the Analyze_Information PIC.

The SSF/CCF collects enough information (e.g., a feature code, prefix, address information, etc.) from the originating access to process the call. This information is collected according to the dialing plan assigned to the originating access:

a) For a call from a non-ISDN line, the SSF/CCF sends an in-band prompt (e.g., dial tone) to prompt the user, attaches a digit receiver, collects digits and applies inter-digit timing. From a non-ISDN line, the "dialing plan in force" is the dialing plan assigned to the line. For example, if a private dialing plan (sometimes referred to as a customized dialing plan) is assigned, then that private dialing plan is considered to be the "dialing plan in force". If, on a given call, the caller dials an access code to escape to a public office dialing plan, then that public office dialing plan is considered to be the "dialing plan in force". In this case, call processing cannot return to the private dialing plan at this SSF/CCF. If a private dialing plan is not assigned, then a public office dialing plan is assumed to be the "dialing plan in force".

b) For a call from an ISDN user following en bloc sending procedures (i.e., all the information necessary to process the call, which can include a feature code, feature activator, *Called Party Number*, etc., is included in the SETUP message), no further action is taken on the originating access interface. For a call from an ISDN user not following en bloc sending procedures, the SSF/CCF prompts the user with a SETUP ACKnowledge message to send more information. Overlap sending procedures are then followed.

   For a call from an ISDN BRI or PRI, the "dialing plan in force" is determined by the type of number and numbering plan field of the *Called Party Number* information element. If this field indicates "unknown" or if the keypad information element is used, then the dialing plan is determined as defined for non-ISDN lines above.

c) For conventional trunks, if required by the trunk type, an appropriate start signal is returned. A digit receiver is attached, digits are collected and inter-digit timing is applied.

   For a call from a conventional trunk, the number received in signaling is expected to be a non-private number that conforms to ITU-T Recommendation E.164.

d) For SS7 trunks, all the information may be available in the IAM or may come in subsequent messages (i.e., overlap sending). A check is made to ensure that the IAM contains all the information necessary to process the call.

   For a call from an SS7 trunk, the number received in signaling is expected to be a non-private number that conforms to E.164.

e) For private-facility trunks, if required by the private-facility trunk group, an appropriate start signal is returned. Depending on the private-facility trunk group, digits may be collected.

Following the trigger detection (due to the DP criteria assigned being met) related to an armed TDP-R or TDP-N in the BCSM at the Collected_Information DP, caused by a call origination attempt, the SSF checks if call gapping, SS7 overload, or service filtering are not in effect for the related call segment. If these conditions are met, then the CollectedInformation operation is invoked by the SSF. The address of the SCF to which the CollectedInformation has to be sent is determined on the basis of trigger instance related data. The SSF provides parameters as per rules defined in 7.4.6. For TDP-R, a control relationship is established to the SCF and the SSF application timer TSSF is set when the SSF sends the CollectedInformation operation for requesting instructions from the SCF. TSSF is used to prevent excessive call suspension time. For EDP-R, TSSF is also used. For TDP-N and EDP-N, no new control relationship is established. Hence, TSSF is not set.

7.4.3.15.3.2 Error handling

If the destination SCF is not accessible, then the call is given final treatment (other treatments are for further study). On expiration of TSSF before receiving any operation, the SSF aborts the interaction with the SCF and the call is given final treatment (e.g., routing to a final announcement). If the calling party abandons after the sending of the CollectedInformation operation, then the SSF aborts the control relationship after the first answer message from the SCF has been received. If the caller abandons the call and TSSF is running, then the transaction ID is held open until TSSF is satisfied or expires. If the caller abandons the call and timer TSSF is not running, then procedures to support the encountered situation (e.g., procedures to report the condition of caller abandon, procedures associated with the condition of TSSF Timer Expired) shall be followed.

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.15.4 Responding entity (SCF)

7.4.3.15.4.1 Normal procedure

SCF preconditions (TDP):

None

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

For TDP-R, on receipt of the CollectedInformation operation, the SCSM moves from “Idle” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked to process the CollectedInformation operation based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For TDP-N, the SCSM remains in state “Idle” and takes appropriate action on the notification.

SCF postconditions (EDP):

For EDP-R, on receipt of the CollectedInformation operation, the SCSM moves from “Waiting for Notification” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF continues. A Service Logic Program Instance (SLPI) is invoked to process the CollectedInformation operation based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For EDP-N, if this is the last event notification and there are no CallInformationReport and ApplyChargingReport pending, the SCSM returns to state “Idle”. Otherwise, the SCSM remains in state “Waiting for Notification or Report”. Appropriate actions on the notification are taken.

7.4.3.15.4.2 Error handling

If the CollectedInformation operation is rejected, then the SCSM remains in the same state. The maintenance function is informed and no SLPI is invoked. Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.16 CollectInformation procedure

7.4.3.16.1 General Description

The CollectInformation is a class 2 operation that is used by the SCF to request the call to return to the Collect_Information PIC, and then perform the basic originating call processing actions associated with this PIC (e.g., the checking of information in the calledPartyNumber parameter with the supported dialing plan). This operation uses only the resources of the SSF/CCF to collect the information. The use of this operation is only appropriate for a call that had not yet left the setup phase.

When the user provides calledPartyNumber, Collect_Information PIC processing includes collecting of destination information from a calling party. When the calledPartyNumber is included (as dialedDigitsCD) in the Collect_Information operation, further collection is not performed (e.g., SSF/CCF checks the provided calledPartyNumber against the supported dialing plan).

7.4.3.16.2 Parameters

– callingPartyNumber *(CallingPartyID)*:
  See T1.113.

– dialedDigitsCD *(CollectedDigits)*:
  This parameter is applied against the supported dialing plan in the SSF/CCF and, if valid, is used in routing of the call. If provided, it replaces the calledPartyNumber for the call.

– dPConverter *(DPConverter)*:
  This parameter instructs the SSF to perform extended DP to DTMF conversion.

– primaryBillingIndicator *(PrimaryBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.

– alternateBillingIndicator *(AlternateBillingIndicator)*:
  This parameter is sent to indicate the AMA call type and service feature identifier for the alternate trunk group.

– secondAlternateBillingIndicator *(SecondAlternateBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the second alternate trunk group.

– overflowBillingIndicator *(OverflowBillingIndicator)*:
This parameter is sent to provide the AMA call type and service feature identifier for the carrier that is used to route the call.

– aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
This parameter is sent to identify an alternate billing number to which the IN service should be billed.

– aMALineNumber *(SEQUENCE OF AMALineNumber)*:
This parameter may include information such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).

– aMAslpID *(AMAslpID)*:
This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.

– aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– serviceProviderID *(ServiceProviderID)*:
This parameter identifies the Message Storage and Retrieval System.

– serviceContext *(ServiceContext)*:
This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).

– aMABillingFeature *(AMABillingFeature)*:
This parameter is reserved for future use.

– aMASequenceNumber *(AMASequenceNumber)*:
This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– alternateDialingPlanInd *(AlternateDialingPlanInd)*:
This parameter is sent to provide alternate dialing plan information to be used by the SSF in assigning a dialing plan to override the current “dialing plan in force”.

– aMAServiceProviderID *(AMAServiceProviderID)*:
This parameter is sent to provide the identifier of a service provider when the service logic resides on an SCP and the service provider is not the company whose equipment is generating the AMA.

7.4.3.16.3 Invoking entity (SCF)

7.4.3.16.3.1 Normal procedure

SCF precondition:

An SLPI has determined that more information from the calling party is required to enable processing to proceed.

SCF postcondition:

SLPI execution is suspended pending receipt of dialed digits.

This operation is invoked in the SCSM FSM state "Preparing SSF Instructions" if the SLP requires additional information to progress the call. It causes a transition of the FSM to the state "Waiting for Notification or Report".

7.4.3.16.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.16.4 Responding entity (SSF)

7.4.3.16.4.1 Normal procedure

As defined for AnalyzeRouteArg with the following exceptions:

CollectedDigits:

When the SSF receives the *CollectedDigits* parameter, the SSF must be able to interpret the digits in this parameter according to the normal dialing plan for the originating non-ISDN line, ISDN interface, or trunk. The *CollectedDigits* parameter contains a string of digits, which should consist of a sequence of dialing plan element(s). For example, a vertical service code plus a directory number gives an example of a sequence of two dialing plan elements as defined below.

The SSF identifies the dialing plan element(s) contained in the *CollectedDigits* parameter until all digits in the *CollectedDigits* parameter have been processed, the dialing plan has been satisfied, or invalid information has been encountered. The SSF disregards any digits beyond those that satisfy the dialing plan in force. If, after processing the operation, the SSF requires further digits from the originating access, then the SSF reverts to normal procedures for prompting the calling user and receiving digits.

Definition of "dialing plan element":

A string of digits that satisfies a dialing plan consists of one or more dialing plan elements. If received from a non-ISDN line or an ISDN interface, dialing plan elements may be separated in time by a prompt (e.g., recall dial tone) to the calling user to request supplementary information. Examples of dialing plan elements include a directory number, a vertical service code, an access code, and a feature activation indicator.

Note that the switch normally performs critical interdigit timing to resolve any ambiguities in the completeness of a dialing plan element. However, no timing information accompanies the digits contained in the *CollectedDigits* parameter. To resolve any ambiguity in the length of a dialing plan element contained in the *CollectedDigits* parameter, the SSF chooses the dialing plan element of maximum length.

Based on the above definition of a dialing plan element, when the SSF receives the *CollectedDigits* parameter, it interprets the digits in the *CollectedDigits* parameter according to the dialing plan in force, as described below.

Definition of "dialing plan in force":

In the context of the CollectInformation operation, the "dialing plan in force" shall be determined as follows:

If the originating access is a non-ISDN line or ISDN interface, and the call has not yet encountered a call forwarding feature within the SSF, then the SSF shall use the dialing plan assigned to the originating access.

If the originating access is an SS7 trunk or a conventional trunk, and the call has not yet encountered a call forwarding feature within the SSF, then the SSF shall use the dialing plan (whether public or private) assigned to the trunk.

If the originating access is a private facility trunk, and the call has not yet encountered a call forwarding feature within the SSF, then the SSF shall use the dialing plan (whether public or private) assigned to the trunk.

If the call has previously encountered a call forwarding feature within the SSF, then the SSF shall use the dialing plan assigned to the directory number/call type that most recently forwarded the call. If no dialing plan is assigned to the directory number/call type, then the SSF shall use the public office dialing plan.

In addition, to determine the dialing plan in force, the SSF shall take into consideration any digits previously received for the call. For example, an access code may cause the dialing plan in force for a call to escape from a private dialing plan to a public office dialing plan.

Identification of a "dialing plan element":

Note that the SSF cannot time the receipt of digits using critical interdigit when it receives digits in the *CollectedDigits* parameter, as it does when it receives digits directly from the originating access. The following rule provides a method for resolving ambiguities in the length of a dialing plan element in the absence of timing information.

To process the digits in the *CollectedDigits* parameter, the SSF shall resolve any ambiguity in the length of a dialing plan element by choosing the longest possible dialing plan element.

CollectedDigits processing:

If the originating access is non-ISDN line or an ISDN interface, and the call has *not* previously encountered a call forwarding feature within the SSF, then the SSF shall proceed as follows:

If the digits in the *CollectedDigits* parameter satisfy the dialing plan in force, then the SSF shall analyze these digits according to the dialing plan in force. Note that these digits shall be subjected to trigger analysis at the Collected_Information DP. If the *CollectedDigits* parameter contains digits beyond those needed to satisfy the dialing plan in force, then the SSF shall ignore the superfluous digits.

If the *CollectedDigits* parameter consists entirely of one or more complete dialing plan elements, but the number of digits in the *CollectedDigits* parameter is *not* sufficient to satisfy the dialing plan in force (i.e., the SSF needs supplemental information to satisfy the dialing plan), then the SSF shall prompt the originating access for supplemental information (e.g., by applying recall dial tone) as if the digit(s) in the *CollectedDigits* parameter had been received from the originating access.

If the information contained in the *CollectedDigits* parameter violates the dialing plan in force, then the SSF shall provide the misdialing treatment appropriate to the originating access. This condition maps to the InvalidInformation event.

If the information contained in the *CollectedDigits* parameter does *not* violate the dialing plan in force, the number of digits is *not* sufficient to satisfy the dialing plan, and the *CollectedDigits* parameter ends with digits that can *not* be identified as a dialing plan element, then the SSF shall provide the treatment appropriate to the originating access had the interdigit timer expired awaiting further digits. This condition maps to the CollectTimeout event.

If the originating access is an SS7 trunk, a conventional trunk or private facility trunk, or if the call has previously encountered a call forwarding feature within the SSF, then the SSF shall proceed as follows:

If the digits in the *CollectedDigits* parameter satisfy the dialing plan in force, then the SSF shall analyze these digits according to the dialing/numbering plan in force. Note that these digits shall be subjected to trigger analysis at the Collected_Information DP. If the *CollectedDigits* parameter contains digits beyond those needed to satisfy the dialing plan in force, then the SSF shall ignore the superfluous digits.

If the information contained in the *CollectedDigits* parameter violates the dialing plan in force, then the SSF shall provide the misdialing treatment appropriate to the originating access. This condition maps to the InvalidInformation event.

If the information contained in the *CollectedDigits* parameter does *not* violate the dialing plan in force, but the number of digits received in the *CollectedDigits* parameter is *not* sufficient to satisfy the dialing plan in force, then the SSF shall provide the treatment appropriate to the originating access had the interdigit timer expired awaiting further digits. This condition maps to the CollectTimeout event.

DPConverter:

If the operation contains the *DPConverter* parameter, and the originating access is a non-ISDN line, then the SSF shall perform extended DP-to-DTMF conversion for digits received from the originating non-ISDN line after the call has been answered.

7.4.3.16.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.17 Continue procedure

7.4.3.17.1 General description

This operation is used to request the SSF to proceed with call processing at the DP at which it previously suspended call processing to await SCF instructions. The SSF continues call processing without substituting new data from the SCF.

7.4.3.17.2 Parameters

- primaryBillingIndicator *(PrimaryBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.
- aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
  This parameter is sent to identify an alternate billing number to which the IN service should be billed.

- aMABusinessCustomerID *(AMABusinessCustomerID)*:
  This parameter is sent to identify the business customer and the type of customer ID.
- aMALineNumber *(SEQUENCE OF AMALineNumber)*:
  This parameter may include information, such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).
- aMAslpID *(AMAslpID)*:
  This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.
- aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
  This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- serviceProviderID *(ServiceProviderID)*:
  This parameter identifies the Message Storage and Retrieval System.
- serviceContext *(ServiceContext)*:
  This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).
- aMABillingFeature *(AMABillingFeature)*:
  This parameter is reserved for future use.
- aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.
- aMAServiceProviderID *(AMAServiceProviderID)*:
  This parameter is sent to provide the identifier of a service provider when the service logic resides on an SCP and the service provider is not the company whose equipment is generating the AMA.

7.4.3.17.3 Invoking entity (SCF)

7.4.3.17.3.1 Normal procedure

SCF precondition:

SCSM is in the state “Preparing SSF instructions”.

SCF postcondition:

SCSM is in the state “Waiting for Notification or Request”, in case monitoring was required, or in the state “Idle”, in case no monitoring was required.

The SCSM is in state “Preparing SSF instructions”. The “Continue” operation is invoked by an SLPI. This causes a SCSM transition to state “Idle” if no subsequent monitoring is required. However, if monitoring is required, such as in the case of armed EDPs or outstanding report requests, the SCSM transitions to state “Waiting for Notification of Report”.

7.4.3.17.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.17.4 Responding entity (SSF)

7.4.3.17.4.1 Normal procedure

The following applies when the SSF receives the Continue operation as a response to an AnalyzedInformation operation:

> The Continue operation requests the SSF to continue call processing for the call (i.e., the SCF does not provide new information for the call). The SSF shall resume processing at the DP where the trigger was detected. The SSF uses the information that it had before querying the SCF to check for other triggers that may apply in order of precedence. If another trigger is detected at the Analyzed_Information DP, the SSF begins a new transaction by sending an AnalyzedInformation operation to the SCF. If a trigger is not detected, then the SSF treats the call as usual.
>
> If the SSF receives the Continue operation after detecting a Specific_Digit_String or N11 trigger, and does not detect another trigger at the Analyzed_Information TDP, it shall route the call as usual. If the SSF receives a Continue operation after any other trigger, it shall route the call if possible. The SSF may not have enough information to route the call. In this case, if the SSF does not detect another trigger, it shall clear the call.

The SSF shall accept the Continue operation after it sends an OCalledPartyBusy operation:

> If Continue is received in response to an OCalledPartyBusy TDP-Request operation, then the SSF shall continue normal call processing and alert the calling party of the busy condition.
>
> If Continue is received in response to an OCalledPartyBusy EDP-Request operation, then the SSF shall continue normal call processing. The SSF shall then detect the O_Called_Party_Busy trigger if it applies.

The SSF shall accept the Continue operation after it sends an ONoAnswer operation:

> If Continue is received in response to an ONoAnswer TDP-Request operation, then the SSF shall continue normal call processing and continue to alert the calling party.
>
> If Continue is received in response to an ONoAnswer EDP-Request operation, then the SSF shall continue normal call processing and detect the O_No_Answer trigger if it applies and the O_No_Answer trigger timer has not yet expired. (If the O_No_Answer trigger timer expires before the Continue operation is received, the SSF shall not encounter the trigger).

The SSF shall accept the Continue operation after it sends a TBusy operation. The SSF shall perform any switch-based terminating services that act on the busy event (e.g., call waiting, call forwarding busy line). If no switch-based terminating features are active then the SSF shall return a busy indication to the originating end of the call.

The SSF shall accept the Continue operation after it sends a TNoAnswer operation:

> The SSF shall continue call processing and continue to alert the called party.

If a TNoAnswer trigger is active for the call, the SSF shall encounter it when the T_No_Answer timer expires. (If the T_No_Answer timer expires before the Continue operation is received, the SSF shall not encounter the trigger).

The SSF shall perform any switch-based terminating services that act on a “no answer” event, such as call forwarding on no answer.

The SSF shall accept the Continue operation after it sends a FacilitySelectedAndAvailable operation. The SSF shall continue call processing and inform the terminating access of the call.

If an event could be detected as both a requested event and a trigger, and if the SSF receives a Continue operation in response to the operation that reported the event as a requested event, then the SSF shall proceed to process the event as a trigger.

7.4.3.17.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.18 EventNotificationCharging procedure

7.4.3.18.1 General description

This operation is used by the SSF to report to the SCF the occurrence of a specific charging event type as requested by the SCF using the “RequestNotificationChargingEvent” operation. The operation supports the options to cope with the interactions concerning the charging.

As several charging events may occur during a connection configuration a possibility exists for the EventNotificationCharging operation to be invoked on multiple occasions. For each connection configuration EventNotificationCharging may be used several times.

7.4.3.18.2 Parameters

– eventTypeCharging *(EventTypeCharging)*:
  This parameter indicates the charging event type that has occurred. Its content is network operator-specific, which may be “charge pulses” or “charge messages”.

– eventSpecificInformationCharging *(EventSpecificInformationCharging)*:
  This parameter contains charging related information specific to the event. Its content is network operator-specific.

– legID *(LegID)*:
  This parameter indicates the leg id on which the charging event type applies.

– extensions *(SEQUENCE OF ExtensionField)*:
  This parameter indicates an extension of an argument data type. Its content is network-specific.

– monitorMode *(MonitorMode)*:
  This parameter indicates how the charging event is reported. When the “monitorMode” is “interrupted”, the event is reported as a request, if the “monitorMode” is “notifyAndContinue the event is reported as a notification. The “monitorMode” “transparent” is not applicable for the EventNotificationCharging operation.

7.4.3.18.3 Invoking entity (SSF)

7.4.3.18.3.1 Normal procedure

SSF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. A charging event has been detected that is requested by the SCF.

SSF postcondition:

No FSM state transition.

The SSF-FSM is in any state except "idle". This operation is invoked if a charging event has been detected that is requested by the SCF. The detected charging event can be caused by: a) another SLPI or b) another exchange. Irrespective of the charging event cause, the SSF performs one of the following actions on occurrence of the charging event (according the corresponding monitorMode):

a) Interrupted:
Notify the SCF of the charging event using "EventNotificationCharging" operation: do not process the event, but discard it. However, call and existing charging processing will not be suspended in the SSF.

b) NotifyAndContinue:
Notify the SCF of the charging event using "EventNotificationCharging", and continue processing the event or signal.

7.4.3.18.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.18.4 Responding entity (SCF)

7.4.3.18.4.1 Normal procedure

SCF precondition:

A "RequestNotificationChargingEvent" has been sent at the request of an SLPI and the SLPI is expecting an "EventNotificationCharging" from the SSF.

SCF postcondition:

No FSM state transition.

On receipt of this operation the SLPI that is expecting this notification can continue. If the corresponding monitor mode was set by the SLPI to Interrupted, the SLPI prepares instructions for the SSF if necessary

7.4.3.18.4.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.19 FacilitySelectedAndAvailable procedure

7.4.3.19.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the Facility_Selected_and_Available DP, or to report an event requested by RequestReportBCMEvent. Refer to 5.7.2 for additional call modeling related semantics.

7.4.3.19.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
If the SSF detects a TDP-Request, the SSF shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the identity of the terminating facility or directory number to which the trigger type is assigned. If the SSF detects a requested event as an EDP-Notification, then the SSF shall send an EDP-Notification message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction.

– bearerCapability (*BearerCapability*):
This parameter indicates the type of bearer capability connection to the user. Refer to the AnalyzedInformation operation for population rules for the bearerCapability parameter.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
If the SSF detects a requested event as an EDP-Notification, then the SSF populates the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of “TRUE.”

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– calledPartyNumber *(CalledPartyID)*:
See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– servingAreaID *(Lata)*:
See 3.1.

– serviceAddressInformationTriggerType *(TriggerCriteriaType):*
This parameter indicates “facilitySelectedAndAvailable”.

– chargeNumber *(ChargeNumber)*:
See 3.1.

– callingPartyNumber *(CallingPartyID)*:
See T1.113 Calling Party Number signaling information. Refer to 7.4.6 for population rules for the callingPartyNumber parameter.

– callingPartysCategory *(ChargePartyStationType)*:
See T1.113 Calling Party’s Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– originalCalledPartyID *(OriginalCalledPartyID)*:
See T1.113 Original Called Number signaling information.

– redirectingPartyID *(RedirectingPartyID)*:
This parameter (if available) is the directory number of the last redirecting party.

– redirectionInformation *(RedirectionInformation)*:
See T1.113 Redirection Information signaling information.

– calledPartysCategory *(CalledPartyStationType)*:
See T1.113 Called Party's Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– sap *(Sap)*:
When the originating access is an SS7 trunk and the received IAM contains the Sap parameter, the SSF shall map the Sap parameter received in the IAM directly to the corresponding sap parameter in the INAP message.

– displayInformationGN *(GenericName)*:
Refer to 7.4.6 for population rules for the displayInformationGN parameter.

– cGEncountered *(ACGEncountered)*:
Refer to 3.1.

– sTRConnection *(STRConnection)*:
This parameter is sent to indicate whether or not the event being reported resulted from a Send_To_Resource operation.

– aMASequenceNumber *(AMASequenceNumber)*:
This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.

7.4.3.19.3 Invoking entity (SSF)

7.4.3.19.3.1 Normal procedures

SSF preconditions:

1. Incoming call received from originating BCSM.
2. An idle facility is selected and available in the terminating BCSM.
3. For a TDP, call gapping or service filtering is not in effect.
4. DP criteria have been met.
5. For a TDP-R or a TDP-N, there is no existing control relationship.
6. For an EDP, there is an existing control relationship and the EDP Facility_Selected_and_Available is armed.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at Facility_Selected_and_Available DP, and a control relationship has been established.
2. For a TDP-N, call processing proceeds to the T_Alerting PIC, and no control relationship has been established.
3. For an EDP, as for the particular event report procedure.

7.4.3.19.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.19.4 Responding entity (SCF)

7.4.3.19.4.1 Normal procedure

SCF preconditions (TDP):

None

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

For a TDP-R, the SCSM moves from "Idle" state to the state "Preparing SSF Instructions". A control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For a TDP-N, the SCSM remains in state "Idle" and takes appropriate action on the notification.

SCF postconditions (EDP):

For an EDP-R, the SCSM moves from "Waiting for Notification" state to the state "Preparing SSF Instructions". A control relationship to the related SSF continues. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For an EDP-N, if this is the last event notification, the SCSM returns to state "Idle". Otherwise, the SCSM remains in state "Waiting for Notification or Report". Appropriate actions on the notification are taken.

7.4.3.19.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.20 ForwardCall procedure

7.4.3.20.1 General description

The ForwardCall operation requests the SSF to forward the call, using the routing information provided. On receipt of a ForwardCall operation, the SSF originates a new call for the forwarding leg and merges the new originating call with the existing terminating call.

7.4.3.20.2 Parameters

– callingPartyNumber *(CallingPartyID)*:
See T1.113.

– chargeNumber *(ChargeNumber)*:
See 3.1.

– callingPartysCategory *(ChargePartyStationType)*:
See T1.113.

– calledPartyNumber *(CalledPartyID)*:
See T1.113. Either the calledPartyNumber or the destinationRoutingAddress shall be provided by the SCF in the ForwardCall operation.

– destinationRoutingAddressON *(OutpulseNumber)*:
This parameter represents the number to be outpulsed.

– travelingClassMark *(Tcm)*:
See 3.1.

– routeListPTG *(PrimaryTrunkGroup)*:
This parameter represents the primary route to be used.

– routeListATG *(AlternateTrunkGroup)*:
This parameter represents the alternate route to be used.

– routeListSATG *(SecondAlternateTrunkGroup)*:
This parameter represents the second alternate route to be used.

– carrier *(Carrier)*:
This parameter represents the primary carrier to be used.

– carrierAC *(AlternateCarrier)*:
This parameter represents the alternate carrier to be used.

– carrierSAC *(SecondAlternateCarrier)*:
This parameter represents the second alternate carrier to be used.

– alertingPattern *(PassiveLegTreatment)*:
See 3.1. This parameter only applies if the network signaling supports this parameter or if SSF is the terminating local exchange for the subscriber.

– primaryBillingIndicator *(PrimaryBillingIndicator)*:
This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.

– alternateBillingIndicator *(AlternateBillingIndicator)*:
This parameter is sent to indicate the AMA call type and service feature identifier for the alternate trunk group.

– secondAlternateBillingIndicator *(SecondAlternateBillingIndicator)*:
This parameter is sent to provide the AMA call type and service feature identifier for the second alternate trunk group.

– overflowBillingIndicator *(OverflowBillingIndicator)*:
This parameter is sent to provide the AMA call type and service feature identifier for the carrier that is used to route the call.

– aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
This parameter is sent to identify an alternate billing number to which the IN service should be billed.

– aMABusinessCustomerID *(AMABusinessCustomerID)*:
This parameter is sent to identify the business customer and the type of customer ID.

– aMALineNumber *(SEQUENCE OF AMALineNumber)*:
This parameter may include information, such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).

– aMAslpID *(AMAslpID)*:
This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.

– aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– serviceProviderID *(ServiceProviderID)*:
This parameter identifies the Message Storage and Retrieval System.

– serviceContext *(ServiceContext)*:
This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).

– aMABillingFeature *(AMABillingFeature)*:
This parameter is reserved for future use.

– aMASequenceNumber *(AMASequenceNumber)*:
This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.

– redirectingPartyID *(RedirectingPartyID)*:
This parameter (if available) is the directory number of the last redirecting party.

– redirectionInformation *(RedirectionInformation)*:
See T1.113 Redirection Information signaling information.

– carrierUsage *(CarrierUsage)*:
This parameter indicates to the SSF how and when to use the carrier information contained in the Carrier, AlternateCarrier and SecondAlternateCarrier parameters.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– aMAServiceProviderID *(AMAServiceProviderID)*:
This parameter is sent to provide the identifier of a service provider when the service logic resides on an SCP and the service provider is not the company whose equipment is generating the AMA.

– destinationRoutingAddress *(DestinationRoutingAddress)*:
See T1.113. Either the calledPartyNumber or the destinationRoutingAddress shall be provided by the SCF in the AnalyzeRoute operation. DestinationRoutingAddress contains a list of CalledPartyIDs.

– destinationRoutingAddressUsage (*DestinationRoutingAddressUsage*):

– This parameter represents the events that shall be monitored by the SSF while processing the *CalledPartyIDs* from the *DestinationRoutingAddress parameter*.

7.4.3.20.3 Invoking entity (SCF)

7.4.3.20.3.1 Normal procedure

The SCF shall populate the parameters and invoke these procedures based on Service Logic.

7.4.3.20.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.20.4 Responding entity (SSF)

7.4.3.20.4.1 Normal procedure

As a result of the merging of the existing terminating call with the new originating call, the terminating call portion is finished in that no more triggers can be detected within the terminating call portion, and the call is treated as an originating call from that point on. The SSF then treats this new originating call portion the same way it would treat the receipt of an AnalyzeRoute operation. Therefore, it is important to note that the ForwardCall operation creates an originating call portion to redirect the call. Additionally, the detectable events after the receipt of the ForwardCall operation are in the originating call portion.

When the SSF receives a ForwardCall operation accompanied by a RequestReportBCMEvent operation, the SSF forwards the call. In addition, all the events requested by the RequestReportBCMEvent apply to this new originating call portion. If the SSF detects a requested event, the SSF sends a message that has the same transaction ID as the RequestReportBCMEvent. The reason for this is that the TerminationAttempt - ForwardCall transaction is still open; in other words, the transaction is extended from the old terminating call portion, which is now dropped, to the new originating call portion. Therefore, the Request or Notification operation has the same transaction ID as the operation that requested the events.

On receiving the ForwardCall operation the SSF shall do the following:

1. Originate a new call for the forwarded leg.
2. Merge the new originating call and the existing terminating call (i.e., bridge or reconfigure the call). The terminating call model is now finished in that while the call is being forwarded, no more terminating triggers within the finished terminating call portion shall be encountered.
3. Route the call as if it received an AnalyzeRoute operation, using the *CalledPartyID, DestinationRoutingAddress*, or *OutpulseNumber* in the ForwardCall operation, as appropriate, to identify the new called party.
4. Process the parameters in ForwardCallArg following the processing rules for the AnalyzeRoute operation.

5. Allow triggers or switch-based features to subsequently be encountered.

When the call is forwarded as a result of a ForwardCall operation, the SSF shall pass *CallingPartyID* from the terminating call portion to the originating call portion and carry it forward with the call.

When the SSF receives a ForwardCall operation that contains the *CallingPartyID* parameter, the SSF shall map the *CallingPartyID* received from the SCF to the:

Calling Party Number information element in the SETUP message if the SSF routes the call to an ISDN line (for both Basic rate and Primary rate ISDN signaling).

Calling Party Number parameter in the IAM if the SSF routes the call to an SS7 trunk (for ISUP signaling).

7.4.3.20.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.21 FurnishChargingInformation procedure

7.4.3.21.1 General description

This operation is used to request the SSF to generate, register a call record or to include some information in the default call record. The registered call record is intended for off-line charging of the call. A possibility exists for the FurnishChargingInformation (FCI) operation to be invoked on multiple occasions. FCI could be applied at the beginning of the call in order to request to start call record generation. In addition FCI can also be applied at the end of the call or connection configuration (e.g., for follow-on calls). In this case FCI is used to include charge related information into the call record that was started at the beginning of the call. When additional FCIs are used it is recommended to arm an EDP-R (indicating the end of call or connection configuration) to be able to apply FCI before the termination of the call record generation.

7.4.3.21.2 Parameters

– fCIBillingChargingCharacteristics:
  This parameter indicates billing and/or charging characteristics. Its content is network operator-specific.

7.4.3.21.3 Invoking entity (SCF)

7.4.3.21.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. An SLPI has determined that a “FurnishChargingInformation” has to be sent by the SCF.

SCF postconditions:

1. No FSM state transition,

2. SLPI execution may continue.

The SCSM FSM is in state “Preparing SSF instruction” or is in state “Queuing FSM”. This operation is invoked by the SCF if an SLPI results in the request of creating a call record to the SSF or to include some billing or charging information into the default call record. In the case of call queuing, this operation may contain information pertaining to the initiation of queuing or the call queuing time duration for call logging purpose. This causes no SCSM FSM state transition.

7.4.3.21.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.21.4 Responding entity (SSF)

7.4.3.21.4.1 Normal procedure

SSF preconditions:

SSF-FSM State c: “Waiting for Instructions” or

SSF-FSM State d: “Waiting for End of User Interaction” or

SSF-FSM State e: “Waiting for End of Temporary Connection” or

SSF-FSM State f: “Monitoring” or

Assisting/hand-off SSF-FSM State b: “Waiting for Instructions”.

SSF postcondition:

No FSM state transition.

On receipt of this operation the SSF performs actions to create the call record according the off line charging scenario that is applicable using the information elements included in the operation:

– registers the complete call record included in the operation;
– generates and registers a call record according the information (charge party, charge level, charge items);
– includes the information received “correlationID” in the default call record that is generated and registered by default at the SSF.

By means of a parameter at the “FurnishChargingInformation” operation the SCF can initiate the pulse metering function of the SSF.

In that case the SSF shall generate meter pulses according to the applicable charging level, account and record them.

The SSF records charge-related data such as the call duration, begin-time stamp or end-time stamp. Additionally the SSF records further data if required.

The charging level can be determined by:

a) the SCF, or
b) the SSF, or

c) a succeeding exchange, or

d) the post processing function.

If a) applies the charging level is included in the “FurnishChargingInformation” operation.

If b) applies the SSF shall determine the charging level based on the corresponding parameters contained in the operation.

If c) applies either the “FurnishChargingInformation” operation contains the corresponding parameters indicating that the charging level shall be determined in a succeeding exchange or the SSF detects during the determination of the charging level based on the provided parameters that the charging level shall be determined in a succeeding exchange.

The SSF can either account received pulses or convert any charging messages received from the B-side to pulses. In both cases, the accumulated pulses are included when the IN call record is generated or ignored.

7.4.3.21.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.22 MonitorForChange procedure

7.4.3.22.1 General description

This operation requests the SSF to monitor a resource for changes in its busy/idle status until the desired status is reached or the monitor request times out.

7.4.3.22.2 Parameters

– resourceStatus *(FacilityStatus)*:
  This parameter indicates the busy/idle status of interest to the SCF.

– monitorDuration *(MonitorTime)*:
  This parameter indicates the length of time that the resource should be monitored (a length of 0 indicates that the SSF should reply immediately with the current status). The SSF shall monitor the line or hunt group until the desired status is reached or the monitor request times out after monitoring for the period of time specified by the monitorDuration parameter.

– calledPartyNumber *(CalledPartyID)*:
  This parameter indicates the *Called Party Number* of the line to be monitored.

– facilityGID *(FacilityGID)*:
  If the calledPartyNumber parameter is not present and the facilityGID is present in the MonitorForChange operation, the SSF shall use the facilityGID parameter to determine the hunt group to be monitored.

– resourceDFGMID *(FacilityMemberID)*:
  This parameter indicates the member of a facility group to be monitored.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the desired type of bearer capability connection to the user. To

determine the busy/idle status the SSF should check whether a call with this type of bearer capability could be established.

- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.22.3 Invoking entity (SCF)

7.4.3.22.3.1 Normal procedure

SCF precondition:

The SCF wishes to monitor the busy/idle status of a line or hunt group at the SSF (a control relationship need not exist between the SCF and the SSF).

SCF postcondition:

SCF waiting for status report.

7.4.3.22.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.22.4 Responding entity (SSF)

7.4.3.22.4.1 Normal procedure

SSF preconditions:

None.

SSF postcondition:

SSF sets timer and monitors resource, sends status report immediately if monitorDuration = 0 or resourceStatus = current status of resource.

7.4.3.22.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.23 MonitorSuccess procedure

7.4.3.23.1 General description

This operation is sent in response to MonitorForChange, if the facility is not in the designated busy/idle status and the monitorDuration was not zero.

7.4.3.23.2 Parameters

– resourceStatus *(FacilityStatus)*:
This parameter indicates the detected status of physical termination resource that is requested to monitor by the SCF.

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

7.4.3.23.3 Invoking entity (SSF)

7.4.3.23.3.1 Normal procedure

SSF precondition:

SSF received MonitorForChange operation from SSF.

SSF postcondition:

a) SSF sent MonitorSuccess to SCF

b) SSF monitors resource, sends status report when timer expires or desired resourceStatus = current status of resource.

7.4.3.23.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.23.4 Responding entity (SCF)

7.4.3.23.4.1 Normal procedure

SCF precondition:

SCF waiting for status report

SCF postcondition:

SCF waiting for status report

7.4.3.23.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.24 NCAData procedure

7.4.3.24.1 General description

To send data from the SCF to the ISDN-connected device or from the ISDN-connected device to the SCF.

7.4.3.24.2 Parameters

– calledPartyNumber: *(CalledPartyID)*
  See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– srhrGroupID *(SrhrGroupID)*:
  This parameter indicates that the contents of the NCAData operation should be broadcast to a group of addresses.

– callingPartyNumber *(CallingPartyID))*:
  See T1.113 Calling Party Number signaling information.

– envelopeEncodingAuthority *(EnvelopeEncodingAuthority)*:
  This parameter is sent to indicate the format and structure of the envelopeContent parameter.

– envelopeContent *(EnvelopContent)*:
  This parameter is defined by the applications (e.g., at the IP and SCP) exchanging information.

– securityEnvelope *(SecurityEnvelope)*:
  This parameter is defined by the applications (e.g., at the IP and SCP) exchanging information.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.24.3 Invoking entity (SCF or SSF)

7.4.3.24.3.1 Normal procedure

SCF or SSF preconditions:

None.

SCF or SSF postcondition:

None.

7.4.3.24.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.24.4 Responding entity (SSF or SCF)

7.4.3.24.4.1 Normal procedure

SSF or SCF preconditions:

None.

SSF or SCF postconditions:

a) SSF processes NCAData and transfers enveloped information to ISDN-connected device; or
b) SCF processes NCAData operation.

7.4.3.24.4.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.25 NCARequest procedure

7.4.3.25.1 General description

To initiate communication between an ISDN-connected device and an SCF.

7.4.3.25.2 Parameters

Invoking Parameters

– calledPartyNumber *(CalledPartyID)*:
See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– callingPartyNumber *(CallingPartyID)*:
See T1.113 Calling Party Number signaling information. Refer to 7.4.6 for population rules for the callingPartyNumber parameter.

– envelopeEncodingAuthority *(EnvelopeEncodingAuthority)*:
This parameter is sent to indicate the format and structure of the envelopeContent parameter.

– envelopeContent *(EnvelopContent)*:
This parameter is defined by the applications (e.g., at the IP and SCP) exchanging information.

– securityEnvelope *(SecurityEnvelope)*:
This parameter is defined by the applications (e.g., at the IP and SCP) exchanging information.

- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

Responding Parameters

- envelopeEncodingAuthority *(EnvelopeEncodingAuthority)*:
  This parameter is sent to indicate the format and structure of the envelopeContent parameter.
- envelopeContent *(EnvelopContent)*:
  This parameter is defined by the applications (e.g., at the IP and SCP) exchanging information.
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.25.3 Invoking entity (SCF or SSF)

The SSF or SCF shall populate the parameters and send this operation based on Service Logic.

7.4.3.25.3.1 Normal procedure

The SSF receives an NCARequest message in a Query Package in a Unidirectional Package from an SCF.

If SSF receives a protocol error (a Reject Component) in response to the NCARequest message, the SSF will include the Problem Code Identifier in the first octet. The second octet will contain the length of two octets. The next two octets will contain the Problem Code received in the Reject component.

If the SSF receives an Abort Package in response to the NCARequest message, the SSF will include either the P-Abort Cause Identifier or the User Abort Identifier in the first octet. The second octet will contain the length of one octet. The third octet will contain the P-Abort Cause obtained from Abort message.

If the CalledPartyID is not valid, then the SSF shall not allow the ISDN-connected device to originate the NCARequest message, and the SSF shall clear the request by sending a RELease COMplete Message.

A Query Package is used to send an NCARequest message from the SSF to an SCF. The SSF sends an NCARequest message when the SSF receives a REGISTER message that contains the envelopNCARequest operation value and the mandatory parameters contained in an FIE.

If ACG Controls do not apply then the SSF shall formulate an NCARequest message in a Query Package. The SSF shall send the NCARequest message to the SCF and start timer T1.

The SSF shall send an NCARequest message in an Invoke (Last) component. The SSF shall also populate the mandatory parameters CalledPartyID, *CallingPartyID*, EnvelopeEncodingAuthority, and the EnvelopContent. These parameters are received in the REGISTER message from the ISDN-connected device. If the optional parameter SecurityEnvelope is present, then the SSF shall also populate this parameter in the NCARequest message.

7.4.3.25.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.25.4 Responding entity (SSF or SCF)

The SSF or SCF shall populate the parameters and send this operation based on Service Logic.

7.4.3.25.4.1 Normal procedure

The SCF would return an NCAResponse message in a Response Package to a previously sent NCARequest message.

The SSF shall accept an NCAResponse message in a Response Package. The NCAResponse message shall be in a Return Result (Last) component correlated to the Invoke component previously sent in the NCARequest message. The SSF shall cancel timer T1.

Upon receipt of a valid NCAResponse message, the SSF shall send a RELease COMplete message to the ISDN-connected device. The SSF shall include a Return Result component in the FIE of the RELease COMplete message. The SSF shall use the Invoke identifier established by the receipt of the REGISTER message. The SSF shall encode the sequence and operation value in the Return Result component in the FIE of the RELease COMplete. The SSF shall include the mandatory parameters EnvelopeEncodingAuthority and EnvelopContent. If the optional parameters SecurityEnvelope is present, then the SSF shall include this parameter in the Result.

An SCF may send an NCARequest message in a Query Package to the SSF. An NCARequest message is a request to initiate communication with an ISDN-connected device and the SSF would send an ISDN NCA signaling message in a REGISTER message to an ISDN-connected device. The SSF would respond to the NCARequest message when it receives a response in a RELease COMplete message from the ISDN-connected device.

The SSF shall accept an NCARequest message in a Query Package. If the SSF receives an NCARequest message in a Query Package for an ISDN-connected device that does not subscribe to “Disallow incoming NCAS messages,” the SSF shall continue to process the message. The NCARequest message shall contain the mandatory parameters CalledPartyID, *CallingPartyID*, EnvelopeEncodingAuthority, and EnvelopContent. The SSF shall expect valid CalledPartyID and the *CallingPartyID* parameters coded as “National number in ISDN numbering plan. If a valid NCARequest message is received, then the SSF shall map the CalledPartyID to CalledDN parameter and use the valid CalledPartyID parameter as an identification of the interface to the ISDN-connected device.

Upon receipt of a valid NCARequest message, the SSF shall determine if it can transfer the information received in the NCARequest message and the required number of octets of the information elements of a REGISTER message to the ISDN-connected device. If the information does not exceed the maximum number of octets that can be sent in an information field of a frame, and the segmentation procedures are not available, then the SSF shall populate a REGISTER message to be sent to the ISDN-connected device if the maximum number of NCA call reference values. The SSF shall include an Invoke component in the FIE. The operation value shall be envelopNCARequest. The SSF shall also include the mandatory parameters CalledPartyID, *CallingPartyID*, EnvelopeEncodingAuthority, and EnvelopContent as arguments. If the optional parameter SecurityEnvelope is present, then the SSF shall also populate this parameter as an argument in the Invoke component. The SSF shall send the REGISTER message to the ISDN-connected device and start timer TINCA.

The SSF initiates the TINCA timer for an ISDN interface after sending a REGISTER (for BRI) or SETUP (for PRI) message to an ISDN-connected device as part of the NCARequest processing. If this timer expires the SSF will notify the SCF that the ISDN-connected device did not respond to the NCARequest message. This timer is applicable to both BRI and PRI ISDN interfaces when an NCARequest message is being processed. The default value for the timer is 30 seconds; the timer range is from 2 to 60 seconds in steps of 1 second. These values are the same as ISDN timer T305.

The SSF is also able to detect unavailability of the IP when sending NCA signaling messages. In this case, the SSF, upon receipt of a NCARequest message from an SCF, sends a REGISTER message (with an envelopNCARequest operation) to the IP. If the timer (TINCA) expires without a response, the SSF sends a Failure_Report message including the *FailureCause* parameter equal to “timerExpired” to the requesting SCF. This action, when successive, is also included in the above “isdnTimeout” and ipTimeout” count.

Each SCF, upon receipt of the NCARequest message, should update the availability information for this IP, and send a response (NCAResponse) back to the IP through the SSF confirming the notification. The SCF should mark all IP/Resource Types associated with this IP as indicated in the EnvelopContent data.

7.4.3.25.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

If an NCARequest message is received in a Query Package and a fatal erroneous data value application error is detected, then the SSF shall report the fatal application error. The SSF shall NOT send the message to the ISDN-connected device.

If the SSF receives a valid NCA_Request message in a Query Package for an ISDN-connected device that subscribes to “Disallow incoming NCAS messages,” then the SSF shall send a Failure_Report message in a Response Package. The SSF shall include the *FailureCause* parameter with a value of “ncasDisallowed” in the Failure_Report message.

Upon receipt of a valid NCARequest message in a Query Package for an ISDN-connected device, the SSF shall determine if the information received in the NCARequest message and the required number of octets of the information elements of a REGISTER message can be transferred by the SSF to the ISDN-connected device. If the information exceeds the maximum number of octets that can be sent in an information field of a frame and the segmentation procedures are not available, then the SSF shall send a Failure_Report message in a Response Package. The SSF shall include the *FailureCause* parameter with a value of “segmentationError” in the Failure_Report message.

If the SSF does not receive an NCAResponse message to the sent NCARequest message and timer T1 expired, then the SSF shall report the error. The SSF shall also clear the request received from the ISDN-connected device by sending a RELease COMplete message. The RELease COMplete message shall contain an FIE with a Return Error component and an error value of “timerExpired.”

If in response to a previously sent NCARequest message, the SSF receives a Return Error component (Failure_Report or an Application_Error message) in a Response Package, the SSF shall send a RELease COMplete message to the ISDN-connected device. The RELease COMplete message shall contain an FIE with a Return Error component. The SSF shall also include either the received *FailureCause* parameter or the ErrorCause parameter as Error parameter to the Return Error component.

If in response to a previously sent NCARequest message, the SSF receives a Return Error component (Failure_Report or an Application_Error message) in a Response Package, the SSF shall send a RELease COMplete message to the ISDN-connected device. The RELease COMplete message shall contain an FIE with a Return Error component. The SSF shall also include either the received *FailureCause* parameter or the ErrorCause parameter as Error parameter to the Return Error component.

If the SSF receives a Conversation with Permission Package in response to the sent NCARequest message, then the SSF shall detect a fatal Unexpected Communication application error. The SSF shall send an Application_Error message in a Response Package. The SSF shall include the ErrorCause parameter with a value of “unexpectedCommunication.” The SSF shall also send a RELease COMplete message. The RELease COMplete message shall contain an FIE with a Return Error component and an error value “proceduralError

If the SSF receives an Abort Package with a Responding Transaction Identifier corresponding to an NCARequest message, the SSF shall send a RELease COMplete message to the ISDN-connected Device. The RELease COMplete message shall contain an FIE with a Return Error component and an error code “abort.” The SSF shall also pass the P-Abort Cause as parameter with the “returnErrorRejectOrAbort” error.

If the SSF receives an NCARequest message in a Query Package and the SSF cannot send a REGISTER message to the ISDN-connected device because the maximum NCA call reference values will be exceeded, then the SSF shall return a Failure_Report message in a Response Package. The SSF shall include the *FailureCause* parameter with a value of “resourceLimitation” in the Failure_Report message.

7.4.3.26 OAnswer procedure

7.4.3.26.1 General description

This operation is sent from the SSF to the SCF at the O_Answer DP, after detecting a valid trigger condition, or to report an event requested by RequestReportBCMEvent.

7.4.3.26.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. This parameter has the same value that was sent with the TDP-Request message that initiated the transaction.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  This parameter indicates that the requested event has been detected as an EDP-Notification.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– localSSPTime *(LocalSSPTime)*:
This parameter is sent to provide SCF a time stamp so that the SCF can calculate the call duration.

7.4.3.26.3 Invoking entity (SSF)

7.4.3.26.3.1 Normal procedure

SSF preconditions (TDP and EDP):

1. Call origination attempt has been initiated.
2. Indication received that the call has been accepted and the terminating party has answered.
3. Call gapping or service filtering are not in effect.
4. DP criteria are met.
5. For a TDP-R, there is no existing control relationship.
6. For an EDP-R, there is an existing control relationship and the EDP O_Answer is armed.
7. For an EDP-N, there is an existing control or monitoring relationship and the EDP O_Answer is armed.

SSF postconditions for TDP:

1. For a TDP-R, basic call processing has been suspended at O_Answer DP, and a control relationship has been established.
2. For a TDP-N, basic call processing proceeds at O_Active PIC, and no control relationship has been established.

SSF postconditions for EDP:

1. The SSF-FSM stays in the state “Monitoring” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested.
2. The SSF-FSM moves to the state “idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” is requested.
3. The SSF-FSM moves to the state “Waiting for Instructions” if the message type was request. Call processing is interrupted.

7.4.3.26.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.26.4 Responding entity (SCF)

7.4.3.26.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate "Waiting for Notification or Request" if the message type was notification and there are still EDPs armed or a "CallInformationReport" or "ApplyChargingReport" requested, or
2. The SCSM-FSM moves to the state "Idle" if the message type was notification and there are no more EDPs armed, no "CallInformationReport" or "ApplyChargingReport" are requested, or
3. The SCSM-FSM moves to the state "Preparing SSF Instructions" if the message type was request.

7.4.3.26.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.27 OCalledPartyBusy procedure

7.4.3.27.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the O_Called_Party_Busy DP, or to report an event requested by RequestReportBCMEvent.

7.4.3.27.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  If the SSF detects a TDP-Request, the SSF shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the identity of the originating facility or directory number to which the trigger type is assigned. If a call previously encountered call forwarding, the serviceAddressInformationServiceKey *(UserID)* is the identity of the last forwarding station. If the SSF detects a requested event as an EDP-Request, then the SSF shall send an EDP-Request message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction.

– bearerCapability (*BearerCapability*):
  See the AnalyzedInformation operation.

– calledPartyNumber *(CalledPartyID):*
  See the AnalyzedInformation operation.

– servingAreaID (*Lata)*:
  See the AnalyzedInformation operation.

– serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
  The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to "oCalledPartyBusy".

– chargeNumber (*ChargeNumber)*:
  See the AnalyzedInformation operation

– callingPartyNumber (*CallingPartyID)*:
  See the AnalyzedInformation operation.

– callingPartysCategory (*ChargePartyStationType):*
  See the AnalyzedInformation operation.

– carrier (*Carrier*):
  See the AnalyzedInformation operation.

– originalCalledPartyID (*OriginalCalledPartyID*):
  See the AnalyzedInformation operation.

– redirectingPartyID (*RedirectingPartyID*):
  See the AnalyzedInformation operation.

– redirectionInformation (*RedirectionInformation*):
  See the AnalyzedInformation operation.

– busyCause *(BusyCause)*:
  If the terminating access is a non-ISDN line, or an ISDN interface for which network-determined user-busy has been detected, the SSF shall populate the busyCause *(BusyCause) p*arameter as indicated in Table 13. The SSF detects network-determined user-busy when the originating call portion has been notified of the detection of the T_Busy event specifying user busy.

Table 13 - Values of BusyCause fields

BusyCause (BusyCause) fields	*value*
Coding Standard	*"CCITT-standard"*
General Location	*"public network serving the remote user"*
Extension Bit	*"diagnostic is not included after octet 2"*
Cause Value Class	*"normal event"*
Cause	*"user busy"*

– If the terminating access is an ISDN interface and user-determined user-busy has been detected, the SSF shall map the received cause information element directly to the busyCause *(BusyCause)* parameter. The SSF detects user-determined user-busy when the originating call portion has been notified of the detection of the CallRejected event specifying user busy.

– If the terminating access is an SS7 trunk, the SSF shall map the received cause indicator parameter in the REL message directly to the busyCause *(BusyCause)* parameter.

The *SSF* shall populate the Cause field of the busyCause *(BusyCause)* parameter with the values indicated in Table 14.

Table 14 - Values of Cause field

Event	*Cause value*
An analog line being out of order	*“Destination out of order”*
An ISDN interface being out of order	*“Destination out of order”*
Maintenance actions	*“Destination out of order”*
directory number not assigned to office equipment	*“No route to destination”*
Termination Denied	*“Call rejected”*
Call rejected event not specifying user busy	*“Call rejected”*

– sap *(Sap)*:
  When the originating access is an SS7 trunk and the received IAM contains the *Sap* parameter, the SSF shall map the *Sap* parameter received in the IAM directly to the corresponding *Sap* parameter in the INAP message.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  If the SSF detects a requested event as a Request, then the SSF shall send a EDP-Request message, and shall populate the serviceAddressInformationMiscCallInfo *(NotificationIndicator) p*arameter with a value of “FALSE.”

– cGEncountered *(ACGEncountered)*:
  See the AnalyzedInformation operation.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– sTRConnection *(STRConnection)*:
  This parameter is sent to indicate whether or not the event being reported resulted from a Send_To_Resource operation.

– aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.27.3 Invoking entity (SSF)

7.4.3.27.3.1 Normal procedures

SSF preconditions:

1. Call origination attempt has been initiated.

2. Indication received from Terminating BCSM that the call is accepted and the terminating party is busy.
3. For a TDP, call gapping or service filtering is not in effect.
4. DP criteria have been met.
5. For a TDP-R or a TDP-N, there is no existing control relationship.
6. For an EDP, there is an existing control relationship and the EDP O_Called_Party_Busy is armed.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at O_Called_Party_Busy DP, and a control relationship has been established.
2. For a TDP-N, call processing proceeds O_Exception, and no control relationship has been established.
3. For an EDP, as for the particular event report procedure.

7.4.3.27.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.27.4 Responding entity (SCF)

7.4.3.27.4.1 Normal procedure

SCF preconditions (TDP):

None

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

For a TDP-R, the SCSM moves from “Idle” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For a TDP-N, the SCSM remains in state “Idle” and takes appropriate action on the notification.

SCF postconditions (EDP):

For an EDP-R, the SCSM moves from “Waiting for Notification” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF continues. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For an EDP-N, if this is the last event notification, the SCSM returns to state “Idle”. Otherwise, the SCSM remains in state “Waiting for Notification or Report”. Appropriate actions on the notification are taken.

7.4.3.27.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.28 ODisconnect procedure

7.4.3.28.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the O_Disconnect DP, or to report an event requested by RequestReportBCMEvent.

7.4.3.28.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  Refer to 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  This parameter indicates that the requested event has been detected as an EDP-Notification.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

– disconnectCause *(DisconnectCause)*:
  This parameter is sent to indicate the specific reason why an O_Disconnect message was sent to the SCF.

– localSSPTime *(LocalSSPTime)*:
  This parameter is sent to provide SCF a time stamp so that the SCF can calculate the call duration.

7.4.3.28.3 Invoking entity (SSF)

7.4.3.28.3.1 Normal procedures

SSF preconditions:

1. Call origination attempt has been initiated
2. Indication received from Terminating BCSM that the call is accepted and the terminating party has answered
3. Disconnect indication received from an originating party, or received from the terminating party via the terminating BCSM
4. For a TDP, call gapping or service filtering is not in effect.
5. DP criteria have been met
6. For a TDP-R or a TDP-N, there is no existing control relationship
7. For an EDP, there is an existing control relationship and the EDP O_Disconnect is armed.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at O_Disconnect DP, and a control relationship has been established
2. For a TDP-N, call processing proceeds to the O_Null & Authorize Termination Attempt PIC, and no control relationship has been established.
3. For an EDP, as for the particular event report procedure.

7.4.3.28.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.28.4 Responding entity (SCF)

7.4.3.28.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.

4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. the SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” are requested, or
3. the SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.28.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.29 ODTMFEntered procedure

7.4.3.29.1 General description

This operation is sent by the SSF to the SCF to report an oDTMFEntered event requested by RequestReportBCMEvent. The SCF can respond with an analyzeRoute, continue, releaseCall, sendToResource, or collectInformation operation.

7.4.3.29.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  Refer to 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  This parameter indicates that the requested event has been detected as an EDP-Request.

– dTMFDigitsDetected *(DTMFDigitsDetected)*:
  This parameter is sent to indicate the actual DTMF digits collected.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

– localSSPTime *(LocalSSPTime)*:
  This parameter is sent to provide SCF a time stamp so that the SCF can calculate the call duration.

7.4.3.29.3 Invoking entity (SSF)

7.4.3.29.3.1 Normal procedures

SSF preconditions:

1. There is an existing control relationship between the SSF and the SCF.
2. The EDP-R oDTMFEntered is armed.

SSF postconditions:

1. Call processing proceeds normally.
2. The SSF is ready to receive SCF responding operation.

Depending on the instructions from the SCF, the oDTMFEntered event is detected only during the state that is specified by the *MonitoringState* parameter from the RequestReportBCMEvent. Based on the *MonitoringState* parameter, the event can be monitored before answer and/or during active state. That is, the event will be detected only if the state that the call is in when the digits are entered matches the state that is specified for monitoring (See RequestReportBCMEvent procedure for *MonitoringState* parameter description).

7.4.3.29.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.29.4 Responding entity (SCF)

7.4.3.29.4.1 Normal procedure

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.

SCF postconditions (EDP):

1. For an EDP-R, the SCSM moves from “Waiting for Notification” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF continues. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

7.4.3.29.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.30 OfferCall procedure

7.4.3.30.1 General description

The OfferCall operation may be sent by the SCF in response to the TBusy operation. The operation asks the SSF to offer the call to the called party (i.e., continue call processing and try to complete the call if possible).

7.4.3.30.2 Parameters

- callingPartyNumber *(CallingPartyID)*:
  See T1.113.
- alertingPattern *(ControllingLegTreatment)*:
  See 3.1. This parameter only applies if the network signaling supports it or if the SSF is the terminating local exchange for the subscriber.
- displayInformation *(DisplayText)*:
  This parameter indicates a text string to be sent to the end-user.
- primaryBillingIndicator *(PrimaryBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.
- aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
  This parameter is sent to identify an alternate billing number to which the IN service should be billed.
- aMABusinessCustomerID *(AMABusinessCustomerID)*:
  This parameter is sent to identify the business customer and the type of customer ID.
- aMALineNumber *(SEQUENCE OF AMALineNumber)*:
  This parameter may include information such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).
- aMAslpID *(AMAslpID)*:
  This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.
- aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
  This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- serviceProviderID *(ServiceProviderID)*:
  This parameter identifies the Message Storage and Retrieval System.
- serviceContext *(ServiceContext)*:
  This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).
- aMABillingFeature *(AMABillingFeature)*:
  This parameter is reserved for future use.

– aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

– aMAServiceProviderID *(AMAServiceProviderID)*:
  This parameter is sent to provide the identifier of a service provider when the service logic resides on an SCP and the service provider is not the company whose equipment is generating the AMA.

7.4.3.30.3 Invoking entity (SCF)

7.4.3.30.3.1 Normal procedure

The SCF shall populate the parameters and invoke these procedures based on Service Logic.

7.4.3.30.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.30.4 Responding entity (SSF)

7.4.3.30.4.1 Normal procedure

The OfferCall operation may contain the *CallingPartyID* parameter. If the *CallingPartyID* parameter is included, that value shall be used as the ISDN Calling Party Number in any ISDN signaling or as the Calling Party Number used by any CLASS feature.The OfferCall operation may contain the *DisplayText* parameter. The *DisplayText* parameter, if included, contains information for display on the called party's CPE. The *DisplayText* parameter can provide one or more fields of display information (e.g., called address information would be provided in one field) to be sent to the called party; the type of information contained in each field is identified by its tag. These tags are a subset of the tags found in the ISDN display text information element.

If the *DisplayText* parameter is included, the SSF should try to deliver this information to the called party. This would allow IN to be involved in a service such as calling name delivery on call waiting. If the called party is served by an ISDN interface, then display procedures similar to those defined for the AuthorizeTermination operation should be used. The OfferCall operation will only be successful if the called party subscribes to call waiting (for a non-ISDN line) or additional call offering (for BRIs).

The SSF should process OfferCallArg as defined for AuthorizeTermination. Information provided in OfferCallArg takes precedence over information previously received.

If the call terminates to an analog line that subscribes to call waiting, and the *DisplayText* parameter is included, the SSF should deliver the information as for the switch-based caller ID on call waiting feature.

If the SSP cannot offer the call to the called party because it is a) an analog line that already has a call waiting; or b) an ISDN line that has no additional call appearances available; then the call shall encounter a Call Forwarding Busy Line feature, if available. If no Call Forwarding Busy Line feature is available, then the SSP shall provide final treatment.

7.4.3.30.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.31 OMidCall procedure

7.4.3.31.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the O_Mid_Call DP, or to report an event requested by RequestReportBCMEvent.

7.4.3.31.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. This parameter identifies the originating facility.

– calledPartyNumber *(CalledPartyID)*:
  See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– callingPartyNumber *(CallingPartyID)*:
  See T1.113 *Called Party Number* signaling information. Refer to 7.4.6 for population rules for this parameter.

– callingPartyBusinessGroupID *(CallingPartyBGID)*:
  See 3.1. The SCF uses this information element to select SLPs based on the group and for authorization purposes.

– calledPartyBusinessGroupID *(CalledPartyBGID)*:
  See 3.1

– callingPartySubaddress *(CallingPartySubaddress)*:
  See T1.113 for subaddress information.

– calledPartySubaddress *(CalledPartySubaddress)*:
  See T1.113 for subaddress information.

– featureRequestIndicator *(FeatureRequestIndicator)*:
  This parameter indicates the type of feature requested.

– carrier *(Carrier)*:
  As defined in 3.1

7.4.3.31.3 Invoking entity (SSF)

7.4.3.31.3.1 Normal procedure

SSF preconditions:

1. Call origination attempt has been initiated.
2. Indication received from Terminating BCSM that the call is accepted and the terminating party has answered.
3. Feature request received from an originating party.
4. For a TDP, code gapping or service filtering are not in effect.

5. DP criteria have been met.
6. For a TDP, there is no existing control relationship.
7. For an EDP, there is an existing control relationship and the EDP O_Mid_Call is armed.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at O_Mid_Call DP, and a control relationship has been established.
2. For a TDP-N, call processing proceeds to the O_Active PIC, and no control relationship has been established.
3. For an EDP-R, basic call processing has been suspended at O_Mid_Call DP, and the existing control relationship continues.
4. For an EDP-N, basic call processing proceeds at the applicable PIC, and the existing relationship continues unless no further EDPs are armed and no “CallInformationReport” or “ApplyChargingReport” is requested.

7.4.3.31.3.2 Error Handling

Generic error handling of the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.31.4 Responding entity (SCF)

7.4.3.31.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. the SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” are requested, or

3. the SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.31.4.2 Error Handling

Generic error handling of the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.32 ONoAnswer procedure

7.4.3.32.1 General description

This operation is sent from the SSF to the SCF at the O_No_Answer DP, after detecting a valid trigger condition, or to report an event requested by RequestReportBCMEvent.

7.4.3.32.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  Refer to 3.1. If the SSF encounters an O_No_Answer trigger, then this parameter identifies the originating facility or directory number to which the trigger is assigned. If a call previously encountered call forwarding, this parameter is the identity of the last forwarding station.
  If the SSF encounters an O_No_Answer requested event, then this parameter has the same value that was sent with the TDP-Request message that initiated the transaction.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– calledPartyNumber *(CalledPartyID)*:
  See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– servingAreaID *(Lata)*:
  See 3.1.

– serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
  This parameter shall be set to “oNoAnswer”.

– chargeNumber *(ChargeNumber)*:
  See 3.1.

– callingPartyNumber *(CallingPartyID)*:
  See T1.113 Calling Party Number signaling information. Refer to 7.4.6 for population rules for the callingPartyNumber parameter.

– callingPartysCategory *(ChargePartyStationType)*:
  See T1.113 Calling Party’s Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– carrier *(Carrier)*:
  As defined in 3.1.

– originalCalledPartyID *(OriginalCalledPartyID)*:
  See T1.113 Original Called Number signaling information.

– redirectingPartyID *(RedirectingPartyID)*:
  This parameter (if available) is the directory number of the last redirecting party.

– redirectionInformation *(RedirectionInformation)*:
  See T1.113 Redirection Information signaling information.

– sap *(Sap)*:
  Refer to 7.4.6 for population rules for the Sap parameter.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  This parameter indicates that a request for instructions has been issued to the SCF.

– cGEncountered *(ACGEncountered)*:
  See 3.1.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– sTRConnection *(STRConnection)*:
  This parameter is sent to indicate whether or not the event being reported resulted from a SendToResource operation.

– aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.32.3 Invoking entity (SSF)

7.4.3.32.3.1 Normal procedure

SSF preconditions (TDP and EDP):

1. Call origination attempt has been initiated.
2. Indication received that the terminating party has not answered within the specified time period.
3. Call gapping or service filtering is not in effect.
4. DP criteria have been met (TDP or EDP).
5. For a TDP-R, there is no existing control relationship.
6. For an EDP-R, there is an existing control relationship and the EDP O_No_Answer is armed.
7. For an EDP-N, there is an existing control or monitoring relationship and the EDP O_No_Answer is armed.

SSF postconditions for TDP:

1. For a TDP-R, basic call processing has been suspended at O_No_Answer DP, and a control relationship has been established.
2. For a TDP-N, default exception handling has been provided, and no control relationship has been established. Use of TDP-N at this DP implies that there is no further alerting of the called party and that switch based no-answer treatments, as applicable, may be invoked.

SSF postconditions for EDP:

1. The SSF-FSM stays in the state “Monitoring” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested.
2. The SSF-FSM moves to the state “idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” is requested.
3. The SSF-FSM moves to the state “Waiting for Instructions” if the message type was request. Call processing is interrupted.

7.4.3.32.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.32.4 Responding entity (SCF)

7.4.3.32.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. the SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” are requested, or
3. the SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.32.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.33 OriginationAttempt procedure

7.4.3.33.1 General description

This operation is sent by the SSF to the SCF at the Origination_Attempt DP, after detecting a valid trigger condition.

7.4.3.33.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
See 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
See T1.113 User Service Information and T1.607 Bearer Capability information element. Refer to 7.4.6 for population rules for the bearer capability parameter.

– chargeNumber *(ChargeNumber)*:
See 3.1. See 7.4.6 for population rules for chargeNumber.

– servingAreaID *(Lata)*:
See 3.1.

– serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
See 3.1. The event applicable to the Origination_Attempt DP is Off_Hook_Immediate.

– callingPartyNumber *(CallingPartyID)*:
See T1.113 Calling Party Number signaling information. Refer to 7.4.6 for population rules for the callingPartyNumber parameter.

– callingPartysCategory *(ChargePartyStationType)*:
See T1.113 Calling Party Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– carrier *(Carrier)*:
See 3.1.

– cGEncountered *(ACGEncountered)*:
See 3.1.

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– sap *(Sap)*:
When the originating access is an SS7 trunk and the received IAM contains the Sap parameter, the SSF shall map the Sap parameter received in the IAM directly to the corresponding sap parameter in the INAP message.

– aMASequenceNumber *(AMASequenceNumber)*:
This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
This parameter indicates that the event is an EDP-N.

7.4.3.33.3 Invoking entity (SSF)

7.4.3.33.3.1 Normal procedure

SSF preconditions (TDP and EDP):

1. Call origination attempt has been initiated
2. Call gapping or service filtering is not in effect
3. DP criteria have been met (TDP or EDP)
4. For a TDP-R, there is no existing control relationship
5. For an EDP-R, there is an existing control relationship and the EDP Origination_Attempt is armed.
6. For an EDP-N, there is an existing control or monitoring relationship and the EDP Origination_Attempt is armed.

SSF postconditions for TDP:

1. For a TDP-R, basic call processing has been suspended at Origination_Attempt DP, and a control relationship has been established
2. For a TDP-N, default exception handling has been provided, and no control relationship has been established.

SSF postconditions for EDP:

1. The SSF-FSM stays in the state “Monitoring” if the message type was notification and there are still EDPs armed.
2. The SSF-FSM moves to the state “idle” if the message type was notification and there are no more EDPs armed.
3. The SSF-FSM moves to the state “Waiting for Instructions” if the message type was request. Call processing is interrupted.

7.4.3.33.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.33.4 Responding entity (SCF)

7.4.3.33.4.1 Normal procedure

SCF preconditions (TDP):

None

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

For a TDP-R, the SCSM moves from “Idle” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For a TDP-N, the SCSM remains in state “Idle” and takes appropriate action on the notification.

SCF postconditions (EDP):

For an EDP-R, the SCSM moves from “Waiting for Notification” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF continues. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For an EDP-N, if this is the last event notification, the SCSM returns to state “Idle”. Otherwise, the SCSM remains in state “Waiting for Notification or Report”. Appropriate actions on the notification are taken.

7.4.3.33.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.34 OSuspended procedure

7.4.3.34.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the O_Suspended DP, or to report an event requested by RequestReportBCMEvent.

7.4.3.34.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  If the SSF detects a requested event as an EDP-Notification, then the SSF shall send an EDP-Notification message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction.

– bearerCapability (*BearerCapability*):
  This parameter indicates the type of bearer capability connection to the user. Refer to the AnalyzedInformation operation for population rules for the bearerCapability parameter.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  If the SSF detects a requested event as an EDP-Notification, then the SSF populates the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of "TRUE."

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.34.3 Invoking entity (SSF)

7.4.3.34.3.1 Normal procedures

SSF preconditions:

1. A call origination attempt has been initiated.
2. An indication has been received from the terminating BCSM that the call is accepted and the called party has answered.
3. A called party disconnect indication has been received.
4. For a TDP, call gapping or service filtering is not in effect.
5. DP criteria have been met.
6. For a TDP-R or a TDP-N, there is no existing control relationship.
7. For an EDP, there is an existing control relationship and the EDP O_Suspended is armed.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at the O_Suspended DP, and a control relationship has been established.
2. For a TDP-N, call processing proceeds to the O_Suspended PIC, and no control relationship has been established.
3. For an EDP, as for the particular event report procedure.

7.4.3.34.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.34.4 Responding entity (SCF)

7.4.3.34.4.1 Normal procedure

SCF preconditions (TDP):

None

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

For a TDP-R, the SCSM moves from “Idle” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For a TDP-N, the SCSM remains in state “Idle” and takes appropriate action on the notification.

SCF postconditions (EDP):

For an EDP-R, the SCSM moves from “Waiting for Notification” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF continues. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For an EDP-N, if this is the last event notification, the SCSM returns to state “Idle”. Otherwise, the SCSM remains in state “Waiting for Notification or Report”. Appropriate actions on the notification are taken.

7.4.3.34.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.35 OTermSeized procedure

7.4.3.35.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the O_Term_Seized DP, or to report an event requested by RequestReportBCMEvent.

7.4.3.35.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  If the SSF detects a requested event as an EDP-Notification, then the SSF shall send an EDP-Notification message, and shall populate the serviceAddressInformationServiceKey *(UserID)*

parameter with the same value that was sent with the TDP-Request message that initiated the transaction.

- bearerCapability (*BearerCapability*):
  This parameter indicates the type of bearer capability connection to the user. Refer to the AnalyzedInformation operation for population rules for the bearerCapability parameter.
- serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  If the SSF detects a requested event as an EDP-Notification, then the SSF populates the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of "TRUE."
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.35.3 Invoking entity (SSF)

7.4.3.35.3.1 Normal procedures

SSF preconditions:

1. A call origination attempt has been initiated.
2. An indication has been received from the terminating BCSM that the called party is being alerted.
3. For a TDP, call gapping or service filtering is not in effect.
4. DP criteria have been met.
5. Or a TDP-R or a TDP-N, there is no existing control relationship.
6. For an EDP, there is an existing control relationship and the EDP O_Term_Seized is armed.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at O_Term_Seized DP, and a control relationship has been established.
2. For a TDP-N, call processing proceeds to the O_Alerting PIC, and no control relationship has been established.
3. For an EDP, as for the particular event report procedure.

7.4.3.35.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.35.4 Responding entity (SCF)

7.4.3.35.4.1 Normal procedure

SCF preconditions (TDP):

None

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

For a TDP-R, the SCSM moves from “Idle” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For a TDP-N, the SCSM remains in state “Idle” and takes appropriate action on the notification.

SCF postconditions (EDP):

For an EDP-R, the SCSM moves from “Waiting for Notification” state to the state “Preparing SSF Instructions”. A control relationship to the related SSF continues. A Service Logic Program Instance (SLPI) is invoked based on the triggerType parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

For an EDP-N, if this is the last event notification, the SCSM returns to state “Idle”. Otherwise, the SCSM remains in state “Waiting for Notification or Report”. Appropriate actions on the notification are taken.

7.4.3.35.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.36 ReleaseCall procedure

7.4.3.36.1 General description

This operation is used to tear down by the SCF an existing call at any phase of the call for all parties involved in the call. This operation may not be sent to an assisting SSF, except in the case of hand-off procedure.

7.4.3.36.2 Parameters

- primaryBillingIndicator *(PrimaryBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.
- aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
  This parameter is sent to identify an alternate billing number to which the IN service should be billed.
- aMABusinessCustomerID *(AMABusinessCustomerID)*:
  This parameter is sent to identify the business customer and the type of customer ID.
- aMALineNumber *(SEQUENCE OF AMALineNumber)*:
  This parameter may include information such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).
- aMAslpID *(AMAslpID)*:
  This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.
- aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
  This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- serviceProviderID *(ServiceProviderID)*:
  This parameter identifies the Message Storage and Retrieval System.
- serviceContext *(ServiceContext)*:
  This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).
- aMABillingFeature *(AMABillingFeature)*:
  This parameter is reserved for future use.
- aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.
- aMAServiceProviderID *(AMAServiceProviderID)*:
  This parameter is sent to provide the identifier of a service provider when the service logic resides on an SCP and the service provider is not the company whose equipment is generating the AMA.

7.4.3.36.3 Invoking entity (SCF)

7.4.3.36.3.1 Normal procedure

SCF precondition:

State 2.1, “Preparing SSF instructions” or State 2.3, “Waiting for Notification or Request”.

SCF postcondition:

State 1, “Idle”, if neither “CallInformationReport” nor “ApplyChargingReport” has to be received from the SSF. All resources (e.g., a queue) related to the call are released by the SCF, or

State 2.3, “Waiting for Notification or Request” if a “CallInformationReport” or “ApplyChargingReport” still has to be received from the SSF.

7.4.3.36.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.36.4 Responding entity (SSF)

7.4.3.36.4.1 Normal procedure

The SSF shall perform the following procedures when it receives a ReleaseCall operation. Procedures differ based on the access type:

- Non-ISDN line, and the ReleaseCall operation is received in response to an operation other than OriginationAttempt: Give dial tone.
- Non-ISDN line, and the ReleaseCall operation is received in response to the OriginationAttempt operation: Wait for caller abandon. If the caller has not disconnected after a suitable period of time, apply reorder tone and await normal disconnect.
- Incoming trunk: The terminating switch shall stop call processing at the TDP where the ReleaseCall is received. If the caller has not disconnected after a suitable period of time, apply reorder tone and await normal disconnect.
- BRI: Apply normal clearing procedures and return channel to idle.
- PRI: Apply normal clearing procedures and return channel to idle.

7.4.3.36.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.37 ReportError procedure

7.4.3.37.1 General description

The SSF and SCF may use the ReportError operation to report the detection of a fatal application error.

7.4.3.37.2 Parameters

– applicationErrorString *(ApplicationErrorString)*:
  This parameter is sent to identify the application error detected and may include information pertaining to the erroneous message.

7.4.3.37.3 Invoking entity (SCF or SSF)

7.4.3.37.3.1 Normal procedure

The SSF sends a ReportError operation to the SCF if a message is not received by the SSF after the SSF sent a message in a TCAP Conversation Package.

The SCF sends a ReportError operation to the SSF if a message is not received by the SCF after the SCF sent a message in a TCAP Conversation Package.

If a fatal application error is reported in a TCAP Unidirectional Package, then a TCAP Invoke(Last) Component with a ReportError operation is sent.

7.4.3.37.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.37.4 Responding entity (SSF or SCF)

7.4.3.37.4.1 Normal procedure

The SSF or SCF should take the appropriate actions for recording the error and providing final treatment for the call.

7.4.3.38 RequestNotificationChargingEvent procedure

7.4.3.38.1 General description

This operation is used to instruct the SSF how to manage the charging events that are received from other FEs not under the control of the service logic instance. The operation supports the options to cope with the interactions concerning the charging. As several connection configurations may be established during a call a possibility exists for the RequestNotificationChargingEvent (RNC) operation to be invoked on multiple occasions. For each connection configuration an RNC may be used several times.

7.4.3.38.2 Parameters

– Sequence of ChargingEvent:
  This parameter contains a list of the charging events and the corresponding monitor types and corresponding legs. For each element in the list the following information elements are included:

  – eventTypeCharging:
    This subparameter indicates the charging event type. Its content is network operator-specific.

  – monitorMode:
    This subparameter indicates the monitorMode applicable for the corresponding

"eventTypeCharging" subparameter. Monitor may be "interrupted", "notifyAndContinue" or "transparent".

– legID:
  This subparameter indicates the leg id of the corresponding event type charging subparameter.

7.4.3.38.3 Invoking entity (SCF)

7.4.3.38.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. An SLPI has determined that a "RequestNotificationChargingEvent" has to be sent by the SCF.

SCF postconditions:

1. No FSM state transition,
2. SLPI execution may continue.

The SCSM FSM is in state "Preparing SSF Instruction" or is in state "Queuing FSM". This operation is invoked by the SCF if an SLPI results in the instruction of SSF how to cope with the interactions concerning the charging. This causes no SCSM FSM state transition.

7.4.3.38.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.38.4 Responding entity (SSF)

7.4.3.38.4.1 Normal procedure

SSF preconditions:

SSF-FSM State c: "Waiting for Instructions", or

SSF-FSM State d: "Waiting for End of User Interaction", or

SSF-FSM State e: "Waiting for End of Temporary Connection", or

SSF-FSM State f: "Monitoring", or

Assisting/Hand-off SSF-FSM State b: "Waiting for Instructions".

SSF postcondition:

No FSM state transition.

On receipt of this operation the SSF performs actions to cope with the interactions concerning the charging according to the information elements included in the operation. The requested charging event can be caused by: a) another SLPI or b) another exchange. Irrespective of by what the charging event is caused the SSF performs one of the following actions on occurrence of the charging event (in accordance with the corresponding monitorMode):

- Interrupted:
  Notify the SCF of the charging event using “EventNotificationCharging” operation, do not process the event or propagate the signal. However call and existing charging processing will not be suspended in the SSF.
- NotifyAndContinue:
  Notify the SCF of the charging event using “EventNotificationCharging”, and continue processing the event or signal without waiting for SCF instructions (handled as for EDP-N for BCSM events).
- Transparent:
  Do not notify the SCF of the event. This ends the monitoring of a previously requested charging event.

Requested charging events are monitored until ended by a transparent monitor mode (or in the case of charging events) until the end of the connection configuration.

In the case that multiple “RequestNotificationChargingEvent” operations are received for the same connection configuration with the same “eventTypeCharging” and “legID”, only the last received “monitorMode” will apply.

7.4.3.38.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.39 RequestReportBCMEvent procedure

7.4.3.39.1 General description

The SCF sends this operation to request a report of subsequent events that may occur on the call. The operation contains a list of requests and notifications that are desired by the SCF. This operation contains a Next Event List (NEL) that is comprised of two types of events: requests and notifications. In addition, this operation may contain values for the O_No_Answer and T_No_Answer timers.

7.4.3.39.2 Parameters

- eDPRequest *(EDPRequest)*:
  The *EDPRequest* parameter contains the list of requests that are requested by the SCF.
- eDPNotification *(EDPNotification)*:
  The *EDPNotification* parameter contains the list of notifications that are requested by the SCF.
- oNoAnswerTimer *(ONoAnswerTimer)*:
  If included, the *ONoAnswerTimer* parameter contains the value for the O_No_Answer timer.
- tNoAnswerTimer *(TNoAnswerTimer)*:
  If included, the *TNoAnswerTimer* parameter contains the value for the T_No_Answer timer.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.
- timeoutTimer *(TimeoutTimer):*
  This parameter contains the value of the SSF timeout timer, which is used to determine the timeout event.

– oDTMFDigitString *(ODTMFDigitString):*
  This parameter contains the specific DTMF digits that the SSF should be monitoring.

– oDTMFNumberOfDigits *(ODTMFNumberOfDigits):*
  This parameter contains the specific number of DTMF digits that the SSF should be monitoring.

– monitoringState *(MonitoringState):*
  This parameter contains the information for the SSF to decide when to monitor the specified ODTMFEntered event.

7.4.3.39.3 Invoking entity (SCF)

7.4.3.39.3.1 Normal procedure

The SSF shall arm the respective EDPs corresponding to the requests that are requested by the SCF in the *EDPRequest* parameter.

The SSF shall arm the respective EDPs corresponding to the notifications that are requested by the SCF in the EDPNotification parameter.

All the requests and notifications received in the RequestReportBCMEvent operation make up a list of requested events requested by the SCF. The SSF stores this list in an event mask called the next event list (NEL).

If the SSF detects a request, then the SSF shall erase the NEL. Thus, the call can encounter subsequent triggers in the call portion. If the SSF detects a notification, then the SSF shall keep the NEL active. In this case, the NEL shall be erased when the SCF sends a Close operation to close the transaction or when the SSF detects a request.

A NEL sent with AnalyzeRoute or ForwardCall may request the following events as notifications:

OTermSeized

OAnswer.

A NEL sent with AnalyzeRoute, ForwardCall, or CollectInformation may request the following events as requests:

OcalledPartyBusy

ODTMFEntered

OnoAnswer

A NEL sent with AnalyzeRoute, OfferCall, or CollectInformation may request the Timeout event as a request.

OSuspended

ODisconnectCalled

NetworkBusy.

A NEL sent with an AnalyzeRoute may request the RouteSelectFailure event as a request.

A NEL sent with Continue (after AnalyzedInformation) may request the following events as requests:

OcalledPartyBusy

ODTMFEntered

ONoAnswer

OSuspended

ODisconnectCalled

NetworkBusy.

A NEL sent with Continue (after AnalyzedInformation) may request the following events as notifications:

OTermSeized

OAnswer.

A NEL sent with AuthorizeTermination may request the following events as notifications:

FacilitySelectedandAvailable

TAnswer.

A NEL sent with AuthorizeTermination may request the following events as requests:

TBusy

TNoAnswer.

A NEL sent with OfferCall or Continue (after TBusy) may request the following events as notifications:

FacilitySelectedandAvailable

TAnswer.

A NEL sent with OfferCall or Continue (after TBusy) may request the TNoAnswer event as a request.

Note that the called party must subscribe to call waiting or Additional Call Offering in order for the SSF to be able to detect TNoAnswer, TAnswer, or FacilitySelectedandAvailable after TBusy.

The RequestReportBCMEvent operation may contain the *ONoAnswerTimer* parameter only if the ONoAnswer request is requested by the SCF, as indicated by the contents of the *EDPRequest* parameter.

The RequestReportBCMEvent operation may contain the *TNoAnswerTimer* parameter only if the TNoAnswer request is requested by the SCF, as indicated by the contents of the *EDPRequest* parameter.

The RequestReportBCMEvent operation may contain the *TimeoutTimer* parameter only if the Timeout request is requested by the SCF, as indicated by the contents of the *EDPRequest* parameter. If the *TimeoutTimer* parameter is included and the Timeout request is requested, the SSF starts the *TimeoutTimer* when the call is answered. If the call is already answered when the request is received, then the *TimeoutTimer* starts immediately. A Timeout EDP-R message is sent to the SCF when the Timeout event is detected while the call is still connected. If the call is disconnected before the Timeout event occurs, the timer is canceled.

The *TimeoutTimer* contains the IntervalTime, which shall be interpreted by the SSP as the time interval in seconds when the Timeout event shall be detected.

The *MonitoringState* parameter contains the following information for the SSF to decide when to monitor the specified ODTMFEntered event.

– Alerting (Y/N): Monitor the oDTMFEntered event immediately after the RequestReportBCMEvent operation is received and before the call is answered.

– Active (Y/N): Monitor the oDTMFEntered event after the call is answered and before the called party goes on-hook.

7.4.3.39.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.39.4 Responding entity (SSF)

7.4.3.39.4.1 Normal procedure

The SSF should arm the appropriate EDP-N's and EDP-R's as indicated. If provided, the *ONoAnswerTimer* or *TNoAnswerTimer* should be set after the appropriate resource has been seized.

7.4.3.39.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.40 ResetTimer procedure

7.4.3.40.1 General description

This class 2 operation is used by the SCF to refresh the TSSF timer in order to avoid the TSSF time-out at the SSF.

7.4.3.40.2 Parameters

– timerID *(TimerID)*:
  This parameter has a default value identifying the T_{SSF} timer.

– timerValue *(TimerValue)*:
  This parameter specifies the value to which the T_{SSF} is to be set.

– extensions *(ExtensionField)*:
  This parameter indicates an extension of an argument data type. Its content is network-specific.

7.4.3.40.3 Invoking entity (SCF)

7.4.3.40.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. An SLPI has determined by the TSCF-SSF guard timer expiration, that the "ResetTimer" operation has to be sent in order to avoid TSSF time-out at the SSF.

SCF postcondition:

The SLPI resets the TSCF-SSF guard timer.

7.4.3.40.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.40.4 Responding entity (SSF)

7.4.3.40.4.1 Normal procedure

SSF preconditions:

1. Call origination attempt has been initiated.
2. Basic call processing has been suspended at a DP.
3. The SSF FSM is in the “Waiting for Instruction” state or in the “Waiting for End of User Interaction” state or in the “Waiting for End of Temporary Connection” state.

NOTE - Whether the TSSF is used or not in the state “Waiting for End of User Interaction” or in the state “Waiting for End of Temporary Connection” is network operator dependent.

SSF postconditions:

1. The TSSF timer has been reset.
2. The SSF FSM remains in the same state.

7.4.3.40.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.41 ResourceClear procedure

7.4.3.41.1 General description

This operation indicates the outcome of a request to route the call to a resource in the SSF Relay case. In the direct SCF to SRF case, this operation does not pass any results back to the SCF. It simply indicates that the bearer connection to the SRF has been disconnected.

7.4.3.41.2 Parameters

– clearCause *(ClearCause)*:
  This parameter indicates why a connection between a user and a resource was terminated.

– dialedDigitsCD *(CollectedDigits)*:
  See 3.1. See 7.4.6 for population rules for dialedDigitsCD.

– dialedDigitsCAI *(CollectedAddressInfo)*:
  See 3.1. See 7.4.6 for population rules for dialedDigitsCAI.

– carrier *(Carrier)*:
  This parameter indicates the carrier selection information and the primary carrier identification to which a call was routed

– failureCause *(FailureCause)*:
  This parameter is sent to indicate that the received operation could not be performed due to the unavailability of a hardware or software resource (i.e., a failure occurred).

– amp1 *(amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– aMAMeasurement *(AMAMeasurement)*:
  This parameter indicates the start date, start time, and total connect time of an interaction between a user and a resource. It also contains a timing guard field that serves as an error check for the time duration field.

– clearCauseData *(ClearCauseData)*:
  This parameter provides information that may be returned from an SRF in cases where the SRF terminates a connection with an ISDN Return Error component.

– iPReturnBlock *(IPReturnBlock)*:
  This parameter contains the result of any user-interaction with the SRF

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.41.3 Invoking entity (SSF)

7.4.3.41.3.1 Normal procedure

SSF preconditions:

1. A control relationship exists between the SCF and the SSF
2. The SSF has established a SendToResource connection with the SRF
3. In the SSF Relay case, the SRF has performed the requested operation and intends to send the results to the SCF.
4. In the direct SCF to SRF case, the SRF has contacted SCF, performed the requested operation, and sent the results to the SCF directly.

SSF postcondition:

None.

The SSF shall populate the parameters and send this operation based on the information obtained from the SRF.

Regardless of where the desired SRF is located, if the SendToResource message is received within a Conversation package, when the SendToResource operation is finished, the SSF returns a ResourceClear message and waits for further information from the SCF.

The ResourceClear message shall never contain both the dialedDigitsCAI (CollectedAddressInfo) and the dialedDigitsCD (*CollectedDigits*) parameters at the same time.

The dialedDigitsCD (*CollectedDigits*) parameter shall contain the digits collected from the caller when a fixed or variable number of digits was requested in the SendToResource message. The SSF shall not include this parameter if the SCF requested "normal number of digits."

The dialedDigitsCAI (CollectedAddressInfo) parameter shall contain the address digits collected from the caller, with the appropriate Nature of Number coding. The SSF shall only include the dialedDigitsCAI (CollectedAddressInfo) parameter if the SCF requested "normal number of digits," or if the SSF is collecting a "fixed number of digits" or a "variable number of digits" from an ISDN BRI user that has sent in response a *Called Party Number* information element and the nature of number and numbering plan is not set to "unknown."

If the SSF detects that invalid information has been received due to the detection of a digit collection timeout, or if an invalid digit string has been received when the SSF is collecting a normal number of digits, the SSF shall send the collected information in the ResourceClear message to the SCF with the *clearCause* set to "invalidCode."

If the caller abandons before the SendToResource is complete, the SSF shall clear the call and send the ResourceClear message to the SCF with the *clearCause* set to "userAbandon."

If the SSF receives a SendToResource message but is not able to process the message because of the unavailability or failure of switch hardware or switch resources, the SSF shall send the ResourceClear message to the SCF with the *failureCause* set to "unavailableResources."

In the direct SCF to SRF case, the SSF receives a disconnect from the SRF and sends the ResourceClear operation to the SCF with a *ClearCause* set to normal.

7.4.3.41.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.41.4 Responding entity (SCF)

7.4.3.41.4.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF
2. The SCF is expecting either a CallInfoFromResource or a ResourceClear from the SSF

SCF postcondition:

1. The SCF shall retire the timer.
2. The SLPI shall be executed to continue its logic.

7.4.3.41.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.42 RouteSelectFailure procedure

7.4.3.42.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the RouteSelectFailure DP, or to report an event requested by RequestReportBCMEvent (NetworkBusy).

7.4.3.42.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:

– See 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– chargeNumber *(ChargeNumber)*:
See 3.1.

– servingAreaID *(Lata)*:
See 3.1.

– serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
This parameter shall be set to “aFR”.

– callingPartyNumber *(CallingPartyID)*:
See T1.113 Calling Party Number signaling information. Refer to 7.4.6 for population rules for the callingPartyNumber parameter.

– callingPartysCategory *(ChargePartyStationType)*:
See T1.113 Calling Party’s Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– calledPartyNumber *(CalledPartyID)*:
See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– travelingClassMark *(Tcm)*:
See 3.1.

– originalCalledPartyID *(OriginalCalledPartyID)*:
See T1.113 Original Called Number signaling information.

– carrier *(Carrier)*:
See 3.1.

– redirectingPartyID *(RedirectingPartyID)*:
This parameter (if available) is the directory number of the last redirecting party.

– redirectionInformation *(RedirectionInformation)*:
See T1.113 Redirection Information signaling information.

– cGEncountered *(ACGEncountered)*:
See 3.1.

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– sap *(Sap)*:
When the originating access is an SS7 trunk and the received IAM contains the Sap parameter, the SSF shall map the Sap parameter received in the IAM directly to the corresponding Sap parameter in the INAP message.

– aMASequenceNumber *(AMASequenceNumber)*:
This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
This parameter indicates that the event is an EDP-N.

7.4.3.42.3 Invoking entity (SSF)

7.4.3.42.3.1 Normal procedure

SSF preconditions:

1. A call origination attempt has been initiated.
2. The *Called Party Number* is available and nature of address has been determined.
3. Call gapping or service filtering is not in effect for the call segment.
4. DP criteria have been met.
5. For a TDP-R, there is no existing control relationship influencing the call segment.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at Route_Select_Failure DP, and a control relationship has been established.
2. For a TDP-N, basic call processing proceeds at O_Exception, and no control relationship has been established.
3. For an EDP, as for the particular event report procedure.

7.4.3.42.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.42.4 Responding entity (SCF)

7.4.3.42.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.

2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. the SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” is requested, or
3. the SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.42.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.43 SendChargingInformation procedure

7.4.3.43.1 General description

This operation is used to instruct the SSF on the charging information to be sent by the SSF. The sending of charging information can either be by charge pulses or signaling or internal if SSF is located in the local exchange. In the local exchange, either charge meter can be updated or a standard call record created. A possibility exists for the SendChargingInformation operation to be invoked on multiple occasions.

NOTE - The interworking between SSF and PSTN is network operator-specific. This operation has many PSTN/IN interactions.

7.4.3.43.2 Parameters

– sCIBillingChargingCharacteristics *(SCIBillingChargingCharacteristics)*:
  This parameter indicates billing and/or charging characteristics. Its content is network operator-specific. The following information element can be included:
  – charge level
– partyToCharge *(PartyToCharge)*:
  This parameter indicates where the charging information must be sent.
– extensions *(SEQUENCE OF ExtensionField)*
  This parameter indicates an extension of an argument data type. Its content is network-specific.

7.4.3.43.3 Invoking entity (SCF)

7.4.3.43.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. An SLPI has determined that a "SendChargingInformation" has to be sent by the SCF.

SCF postconditions:

1. No FSM state transition,
2. SLPI execution may continue.

The SCSM FSM is in state "Preparing SSF Instruction" or is in state "Queuing FSM". The SendChargingInformation procedure shall be invoked by the SCF in accordance with the demands of the SLPI for relevant charging information. If appropriate this information shall be sent back down the call path.

This causes no SCSM FSM state transition.

7.4.3.43.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.43.4 Responding entity (SSF)

7.4.3.43.4.1 Normal procedure

SSF preconditions:

SSF-FSM State c: "Waiting for Instructions", or

SSF-FSM State d: "Waiting for End of User Interaction", or

SSF-FSM State e: "Waiting for End of Temporary Connection", or

SSF-FSM State f: "Monitoring", or

Assisting/hand-off SSF-FSM State b: "Waiting for Instructions".

SSF postcondition:

No FSM state transition.

On receipt of this operation the SSF performs actions to send the charging information. The sending of charging information can either be by charge pulses or signaling or internal if SSF is located in local exchange. In the local exchange, either charge meter can be updated or a standard call record created. The interworking between SSF and PSTN is network operator-specific. This operation has many PSTN/IN interactions:

For instance, by sending an operation "SendChargingInformation" the SCF instructs the SSF to initiate the PSTN/ISDN charging functions according to the given information about the charging level to use.

The charging level can be determined either by one of the following functions

a) the SCF, or

b) the SSF, or

c) the charging function in a succeeding exchange.

In case a) the SCF has determined the charging level the “SendChargingInformation” operation contains the charging level to be applied.

In case b) the SSF determines the charging level the “SendChargingInformation” operation contains the parameters to determine the charging level.

If the charging level was determined by the IN (SCF or SSF) the SSF provides the charging level to be applied to the PSTN/ISDN charging functions (cases a) and b)).

In case c) the charging level is determined in a succeeding exchange. The “SendChargingInformation” operation either contains the corresponding parameters indicating this fact or the SSF detects during trying to determine the charging level based on SCF provided parameters that the charging level shall be determined in a succeeding exchange. Based on already existing PSTN/ISDN capabilities the SSF provides the PSTN/ISDN charging functions with the necessary information, and backward charge messages shall be transferred down the call path when allowed by the SCF (generated by a succeeding exchange, for example an international gateway).

In the scenario described above charging/billing is performed by means of existing mechanisms of the PSTN/ISDN initiated and controlled by the IN.

That means the determination of the charging method - on-line or off-line - and the items to be charged for shall be done in the basic network, just as for the charge generation and the charge registration.

7.4.3.43.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.44 SendNotification procedure

7.4.3.44.1 General description

This operation is sent by the SCF to request the SSF to send a notification when the call exits to the NULL PIC because it disconnected, cleared, or could not be completed.

7.4.3.44.2 Parameters

– correlationID *(EchoData)*:
  This parameter correlates the SendNotification request with the subsequent TerminationNotification.

7.4.3.44.3 Invoking entity (SCF)

7.4.3.44.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF.
2. The service logic wishes to receive notification of the call termination.

SCF postcondition:

The SCF is waiting to receive a TerminationNotification.

7.4.3.44.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.44.4 Responding entity (SSF)

7.4.3.44.4.1 Normal procedure

SSF preconditions:

1. Call origination attempt has been initiated.
2. Basic call processing has been suspended at a DP.
3. The SSF waits for instructions.

SSF postconditions:

1. The SSF performs the call processing actions requested by the SCF.
2. The SSF waits for call termination (call model returns to NULL PIC).
3. Upon call termination, the SSF sends TerminationNotification to the SCF with call information.

7.4.3.44.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.45 SendToResource procedure

7.4.3.45.1 General description

The SendToResource operation requests the SSF to connect the calling party to an SRF. The parameters of the SendToResourceArg specify the particular function the SRF should provide. SSF/SRF processing of each parameter is described below. Note that if the SendToResource operation does not include a *DestinationAddress*, then the SRF is assumed to be physically located in the switch.

In addition, the SendToResource operation also supports the direct SCF to SRF infromation transfer case by including a correlationBlock in the strParameterBlock and setting the resourceType to correlationBlock.

7.4.3.45.2 Parameters

- resourceType *(ResourceType)*:
  The *resourceType* parameter identifies the specific resource type (or capability) that needs to be provided to the user.
- strParameterBlock *(StrParameterBlock)*:
  The *strParameterBlock* parameter is sent to provide any information needed for the capability specified by the *resourceType* parameter. Therefore, it can take one of a number of different codings depending on the capability that is to be utilized. Each resource type value implies its own parameter block and its own coding. Examples include “Play Announcement” and “Play Announcement and Collect Digits.” Information collected from a user is returned to the SCF in a ResourceClear operation. When a *correlationBlock* is used, the SendToResource operation is supporting the direct SCF to SRF information transfer case. The *correlationBlock* contains the *sCFCorrelationID* and *sCFID*. Based on the information from the *correlationBlock*, the SRF will establish the direct relation with SCF.
- disconnectFlag *(DisconnectFlag)*:
  The presence of the *disconnectFlag* parameter indicates that the SSF should disconnect the call party after the Send_To_Resource interaction is complete. If the *disconnectFlag* parameter is not present, the SSF should not disconnect the call party.
- answerIndicator *(AnswerIndicator)*:
  The presence of the *answerIndicator* parameter indicates that the SSF should return answer supervision to the caller when the caller is connected during a Send_To_Resource interaction. If the *answerIndicator* parameter is not present, the SSF should not return answer supervision.
- primaryBillingIndicator *(PrimaryBillingIndicator)*:
  This parameter is sent to provide the AMA call type and service feature identifier for the primary trunk group and for services when the primary trunk group is not provided.
- aMAAlternateBillingNumber *(AMAAlternateBillingNumber)*:
  This parameter is sent to identify an alternate billing number to which the IN service should be billed.
- aMABusinessCustomerID *(AMABusinessCustomerID)*:
  This parameter is sent to identify the business customer and the type of customer ID.
- aMALineNumber *(SEQUENCE OF AMALineNumber)*:
  This parameter may include information, such as the calling party ID, incoming terminating number, or Automatic Number Identification (ANI).
- aMAslpID *(AMAslpID)*:
  This parameter is sent to indicate that the SSF should override normal switch-based recording and invoke IN AMA record generation. The AMAslpID identifies a service or a unique subset of service functionality.
- aMADigitsDialedWC *(SEQUENCE OF AMADigitsDialedWC)*:
  This parameter is sent to provide any digit string that the customer dialed along with a context identifier to indicate the name of the digit string (e.g., Customer Dialed Account Recording [CDAR], Access Code, and Authorization Code).
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- iPRoutingAddress *(DestinationAddress)*:
  This parameter provides the address of an SRF resource.

- dPConverter *(DPConverter)*:
  This parameter instructs the SSF to perform extended DP to DTMF conversion.
- aMAMeasure *(AMAMeasure)*:
  This parameter is sent to indicate whether or not a duration time measurement should be made. This parameter also indicates the destination of the duration time that is measured.
- serviceProviderID *(ServiceProviderID)*:
  This parameter identifies the Message Storage and Retrieval System.
- serviceContext *(ServiceContext)*:
  This parameter is sent to provide the identification of the context of a service (e.g., service identification, feature identification).
- aMABillingFeature *(AMABillingFeature)*:
  This parameter is reserved for future use.
- aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which IN triggers are invoked within the context of an originating or terminating call portion.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.45.3 Invoking entity (SCF)

7.4.3.45.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and SSF
2. The SLPI intends to connect one of the call parties to an SRF

SCF postcondition:

The SCF shall start a timer and wait for a response to be returned from the SSF

The SCF shall populate the parameters and send this operation based on Service Logic.

7.4.3.45.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.45.4 Responding entity (SSF)

7.4.3.45.4.1 Normal procedure

SSF preconditions:

A control relationship exists between the SCF and the SSF

SSF postconditions:

The SSF shall establish a SendToResource connection with the SRF

Within a SendToResource message, the *DestinationAddress* parameter identifies the location of the desired resource. The SSF interprets a SendToResource message without a *DestinationAddress* as a request to access an internal switch resource. A SendToResource message containing a *DestinationAddress* parameter is interpreted as a request to access a resource located either at an SRF directly connected to the SSF (i.e., at a local SRF) or at an SRF directly connected to another SSF (i.e., at a remote SRF).

Regardless of where the desired SRF is located, if the SendToResource message is received within a Conversation package, when the SendToResource operation is finished, the SSF returns a ResourceClear message and waits for further information from the SCF.

If a SendToResource message requesting an internal SRF is received within a Response Package, the SSF closes the SSF-SCF transaction and performs the SendToResource operation. In this case, the SSF does not return a ResourceClear message to the SCF. The SSF treats a SendToResource message requesting a resource at a local or remote SRF as an error if received within a Response Package.

If the SendToResource message requests the use of an SRF directly connected to the SSF (i.e., at a local SRF), then the SSF uses messages on an ISDN D channel to set up a bearer channel between the SSF and SRF, and to carry IN service-related information to the SRF. The SSF sends an Invoke component that contains an operation “sendToIPResource” in a Facility Information Element (FIE) to the SRF. The FIE is carried in a FACILITY or a SETUP message and the FIE may be repeated in a given message. The response from the SRF is returned in an FIE carried in a FACILITY, DISConnect, or a RELease COMplete message. Along with the “sendToIPResource” operation, other operations can also be carried in separate FIEs in the same message. For the “sendToIPResource” operation, ALERTing, CALL PROCeeding, RELease, and CONNect messages are not allowed to carry the FIE.

If the SendToResource message requests the use of an SRF directly connected to another SSF (i.e., at a remote SRF), the local SSF sends an ISDNUP Initial Address Message (IAM) to establish a trunk connection to the remote SSF to which the remote SRF is connected. The contents of the SendToResource message’s *DestinationAddress* parameter is used as the IAM’s *Called Party Number* parameter. This IAM contains a Remote Operations (RO) parameter populated with information from the SendToResource message. Then, in the same way that a local SSF interacts with a local SRF, the remote SSF initiates call setup to the remote SRF and requests the SRF to perform the necessary interaction with the user. Once the connection to the remote SRF is established, the user and remote SRF can interact. After the interaction with the user is completed, the remote SRF sends the remote SSF any information that the user may have provided. If only a single interaction is required, or if this is the final interaction for the call, the remote SRF passes the collected information to the remote SSF in an FIE of an ISDN access DISConnect message. The remote SSF passes this information to the local SSF in an ISDNUP Release (REL) message, with the RO parameter containing the collected information. Finally, the local SSF passes the information to the SCF in a ResourceClear message.

If multiple interactions are required, the remote SRF passes intermediate collected information to the remote SSF in the FIE of an ISDN access FACILITY message. The remote SSF passes this intermediate information to the local SSF in the RO parameter of an ISDNUP Facility (FAC) message. The local SSF passes the intermediate information to the SCF in a CallInfoFromResource message. In response, the SCF normally sends a CallInfoFromResourceRR message to the local SSF. The local SSF sends the remote SSF a FAC message, with the RO parameter containing the information from the CallInfoFromResourceRR message. The remote SSF passes the information to the remote SRF in the FIE of an ISDN access FACILITY message. If another interaction occurs, the remote SRF again passes the collected information to the SCF as previously described.

If the correlationBlock is contained in the message, direct SCF to SRF communication is being requested by the SCF. The SSF connects the call to the SRF and treats the CorrelationBlock the same way as a FlexParameterBlock. This means that the CorrelationBlock is passed to the SRF using the same method that carries the FlexParameterBlock. Depending on the type of the connection between the SSF and the

SRF, the CorrelationBlock may be carried within a Facility Information Element (FIE) (for DSS1) or a Remote Operations (RO) parameter (for ISUP)

There are two parameters inside the CorrelationBlock.

– sCFCorrelationID: This is the ID used by the SCF to correlate the messages from the SRF. Since the correlationID is assigned and matched by the same SCF, it can be in any format as long as it is non-ambiguous to the SCF.

– scfID: This is the parameter that identifies the address of the sending SCF. The SRF uses it to send messages to the SCF directly.

Based on the scfID, the SRF sends an AssistRequestInstructions message, which contains the correlationID, directly to the SCF for instructions and establishes the direct SCF-SRF relation.

The SCF then responds through the direct connection with an InstructionsToSRF message to instruct the SRF to perform certain functions, such as play announcement and/or collect digits. The SRF uses CallInfoFromResource to pass the result of any performed function back to the SCF. In addition, the SCF can use the CallInfoFromResourceRR message to send more instructions to the SRF or just set the disconnectFromIPFrobidden parameter to FALSE so that the SRF will disconnect the bearer channel to the SSF. Once the bearer channel is disconnected, the SSF will then send a ResourceClear message to the SCF with *ClearCause* set to normal.

Alternatively, the SCF can send a CancelResourceEvent message to the SSF to disconnect the bearer channel to the SRF.

7.4.3.45.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.46 ServiceFilteringResponse procedure

7.4.3.46.1 General description

This operation is used to report the values of counters specified in a previous sent “ActivateServiceFiltering” operation to the SCF.

7.4.3.46.2 Parameters

– countersValue *(CountersValue)*:
The parameter contains the count of calls filtered during the filtering period. It is a list of counter identifications and the related values.

– filteringCriteria *(FilteringCriteria)*:
This parameter is used to address the concerned service logic at the SCF.

– extensions *(SEQUENCE OF ExtensionField)*:
This parameter indicates an extension of an argument data type. Its content is network-specific.

– responseCondition *(ResponseCondition)*:
This parameter is used to identify the reason why the ServiceFilteringResponse is sent:

– intermediateResponse indicates that service filtering is active, a call is received and the interval timer is expired, or that service filtering is active and the threshold value, “numberOfCalls”, is reached.

– lastResponse indicates that the duration time is expired and service filtering is stopped, or that the stop time is met and service filtering is stopped.

7.4.3.46.3 Invoking entity (SSF)

7.4.3.46.3.1 Normal procedure

SSF preconditions:

1. Service filtering is running and the interval time is expired and a call is received, or
2. Service filtering is running and the threshold value is reached, or
3. Service filtering has been finished (duration time expired or stop time met), or
4. The operation “ActivateServiceFiltering” is received and encounters an active service filtering entity.

SSF postcondition:

Service filtering proceeds or is ended depending on the duration time.

The SSF sends the “ServiceFilteringResponse” operation to the SCF. The “filteringCriteria” parameter is provided to enable the addressing of the concerned service logic at the SCF.

Before “ServiceFilteringResponse” is sent, the SSF checks whether call gapping criteria are met. If so, the “ServiceFilteringResponse” is not sent and the counting continues without resetting the counters. The last “ServiceFilteringResponse” (stop time is met or duration time expired) is sent without checking any call gap criteria.

After “ServiceFilteringResponse” is sent, the service filtering counters are reset.

If service filtering proceeds after sending “ServiceFilteringResponse” (e.g., interval time expired) then the SSME-FSM remains in the state “Non-Call Associated Treatment”.

If service filtering is stopped after sending “ServiceFilteringResponse” (duration time expired or stop time is met) then the SSME-FSM moves to the “Idle Management” state. All allocated resources are released, i.e., the SSME-FSM is removed as well.

7.4.3.46.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.46.4 Responding entity (SCF)

7.4.3.46.4.1 Normal procedure

SCF preconditions:

1. Service filtering is running.

2. The SCME is in the state “Waiting for Service Filtering Response”.

SCF postcondition:

The SCME forwards the received counter values to the SLPI.

The operation is handled by the Service Filtering FSM part of the SCF Management Entity (SCME). The SCME passes the received counter values to the SLPI where they are added to previously received counter values.

The “filteringCriteria” parameter as provided in “ServiceFilteringResponse” is used to address the SCME and the concerned service logic instance.

The Service Filtering FSM of the SCME remains in the state “Waiting For SSF Service Filtering Response” until the internal service filtering duration time in the SLPI expires. Then the SLPI informs the SCME about timer expiration. Now the SCME moves to the state “Service Filtering Idle”.

7.4.3.46.4.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.47 StatusReported procedure

7.4.3.47.1 General description

This operation indicates the outcome of a request to monitor a resource for changes in its busy/idle status.

7.4.3.47.2 Parameters

– resourceStatus *(FacilityStatus)*:
  This parameter indicates the current busy/idle status of the resource.

– reportCondition *(StatusCause)*:
  This parameter indicates the cause for sending the report. It may be “statusMatch” if the facility reaches the requested busy/idle status before the monitor times out, or “timeOut” if the timer expires before the facility reaches the requested busy/idle status.

– failureCause *(FailureCause)*:
  If the facility to be monitored could not be monitored (e.g., because the facility does not exist, or too many monitor requests are in effect), the SSF immediately sends a StatusReported message with this parameter set to “unavailableResources”.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.47.3 Invoking entity (SSF)

7.4.3.47.3.1 Normal procedure

SSF precondition:

> SSF received MonitorForChange operation from SSF. If monitorDuration = 0 or the resource is in the desired status, then the SSF sends StatusReported immediately. Otherwise, the SSF sets a timer and sends StatusReported when the resource reaches the desired status or the timer expires.

SSF postcondition:

> SSF sent status report when timer expires or desired resourceStatus = current status of resource.

7.4.3.47.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.47.4 Responding entity (SCF)

7.4.3.47.4.1 Normal procedure

SCF precondition:

> SCF waiting for status report.

SCF postcondition:

> SCF receives status report.

7.4.3.47.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.48 TAnswer procedure

7.4.3.48.1 General description

This operation is sent from the SSF to the SCF at the T_Answer DP, after detecting a valid trigger condition, or to report an event requested by RequestReportBCMEvent.

7.4.3.48.2 Parameters

- serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. This parameter has the same value that was sent with the TDP-Request message that initiated the transaction.
- bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.
- serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  This parameter indicates that the requested event has been detected as an EDP-Notification.
- amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.
- amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.
- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.48.3 Invoking entity (SSF)

7.4.3.48.3.1 Normal procedure

SSF preconditions (TDP):

1. An incoming call has been received from the originating BCSM
2. A call has been accepted and the terminating party has answered
3. For TDP, call gapping or service filtering are not in effect for the call segment
4. DP criteria have been met
5. For a TDP-R, there is no existing control relationship influencing the call segment

SSF preconditions (EDP):

1. For an EDP-R, there is an existing control relationship and the EDP T_Answer is armed.
2. For an EDP-N, there is an existing monitoring or control relationship and the EDP T_Answer is armed.

SSF postconditions (TDP):

1. For a TDP-R, basic call processing has been suspended at T_Answer DP, and a control relationship has been established
2. For a TDP-N, basic call processing proceeds at T_Answer PIC, and no control relationship has been established

SSF postconditions (EDP):

1. The SSF-FSM stays in the state “Monitoring” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested.
2. The SSF-FSM moves to the state “idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” is requested.

3. The SSF-FSM moves to the state “Waiting for Instructions” if the message type was request. Call processing is interrupted.

7.4.3.48.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.48.4 Responding entity (SCF)

7.4.3.48.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. The SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” are requested, or
3. The SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.48.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.49 TBusy procedure

7.4.3.49.1 General description

This operation is sent by the SSF to the SCF after detecting a valid trigger condition at the T_Busy DP, or to report an event requested by RequestReportBCMEvent.

7.4.3.49.2 Parameters

When the SSF sends this information to the SCF, it should use the same rules as for TerminationAttemptArg, with the following exceptions:

– serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. When the SSF encounters a T_Busy trigger, this parameter identifies the terminating facility or directory number to which the trigger is assigned. When the SSF encounters a T_Busy requested event, this parameter has the same value that was sent with the TDP-Request message that initiated the transaction.

– serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
  This parameter indicates "tBusy".

– busyCause *(BusyCause)*:
  See 3.1. See 7.4.6 for population rules.

– busyType *(BusyType)*:
  Indicates whether the SSF is able to accept an OfferCall operation. Refer to 7.4.6 for population rules.

– serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  This parameter indicates that a request for instructions has been issued to the SCF.

– sap *(Sap)*:
  Refer to 7.4.6 for population rules for the Sap parameter.

7.4.3.49.3 Invoking entity (SSF)

7.4.3.49.3.1 Normal procedure

SSF preconditions:

1. A call termination attempt has been initiated.
2. The *Called Party Number* is available and the nature of address has been determined.
3. Call gapping or service filtering is not in effect for the call segment.
4. DP criteria have been met.
5. For a TDP-R, there is no existing control relationship influencing the call segment.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at the T_Busy DP, and a control relationship has been established.
2. For a TDP-N, basic call processing proceeds at T_Exception, and no control relationship has been established.
3. For an EDP, as for the particular event report procedure.

The SSF has sufficient information available associated with the terminating call portion. The SSF shall detect T_Busy when the terminating access is network-determined user-busy. The conditions that result in the detection of network-determined user-busy depend on the type of terminating access and subscribed services (an analog line access, DSS1, multiline hunt group, etc.).

7.4.3.49.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.49.4 Responding entity (SCF)

7.4.3.49.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate "Waiting for Notification or Request" if the message type was notification and there are still EDPs armed or a "CallInformationReport" or "ApplyChargingReport" requested, or
2. The SCSM-FSM moves to the state "Idle" if the message type was notification and there are no more EDPs armed, no "CallInformationReport" or "ApplyChargingReport" are requested, or
3. The SCSM-FSM moves to the state "Preparing SSF Instructions" if the message type was request.

7.4.3.49.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.50 TerminationAttempt procedure

7.4.3.50.1 General description

This operation is sent from the SSF to the SCF at the Termination_Attempt DP, after detecting a valid trigger condition.

7.4.3.50.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
See 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– calledPartyNumber *(CalledPartyID)*:
See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– servingAreaID *(Lata)*:
See 3.1.

– serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
This parameter indicates “terminationAttempt”.

– calledPartysCategory *(CalledPartyStationType)*:
See T1.113 Called Party’s Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– chargeNumber *(ChargeNumber)*:
See 3.1.

– callingPartyNumber *(CallingPartyID)*:
See T1.113 Calling Party Number signaling information. Refer to 7.4.6 for population rules for the callingPartyNumber parameter.

– callingPartysCategory *(ChargePartyStationType)*:
See T1.113 Calling Party’s Category signaling information. Refer to 7.4.6 for population rules for the callingPartysCategory parameter.

– travelingClassMark *(Tcm)*:
See 3.1.

– originalCalledPartyID *(OriginalCalledPartyID)*:
See T1.113 Original Called Number signaling information.

– redirectingPartyID *(RedirectingPartyID)*:
This parameter (if available) is the directory number of the last redirecting party.

– redirectionInformation *(RedirectionInformation)*:
See T1.113 Redirection Information signaling information.

– displayInformationGN *(GenericName)*:
This parameter indicates the generic name of the originating party.

– cGEncountered *(ACGEncountered)*:
See 3.1.

– amp1 *(Amp1)*:
This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– sap *(Sap)*:
When the originating access is an SS7 trunk and the received IAM contains the Sap parameter, the SSF shall map the Sap parameter received in the IAM directly to the corresponding Sap parameter in the INAP message.

– sTRConnection *(STRConnection)*:
  This parameter is sent to indicate whether or not the event being reported resulted from a Send_To_Resource operation.

– aMASequenceNumber *(AMASequenceNumber)*:
  This parameter is sent to indicate the order in which triggers are invoked within the context of an originating or terminating call portion.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

– genericDigitsList *(GenericDigitsList):*

– This parameter includes a GenericDigits parameter that contains signaled Location Information for a wireline Emergency Caller or cell site and sector information for a wireless Emergency Caller (See T1.628 for Emergency Calling Service description and usage).

– callingGeodeticLocation *(CallingGeodeticLocation):*
  This parameter contains the latitude and longitude information of an Emergency Caller (See T1.628 for Emergency Calling Service description and usage).

7.4.3.50.3 Invoking entity (SSF)

7.4.3.50.3.1 Normal procedure

SSF preconditions:

1. An incoming call has been received.
2. Call gapping or service filtering are not in effect.
3. TDP criteria have been met.
4. For a TDP-R there is no existing control relationship.

SSF postconditions:

For a TDP-R, basic call processing has been suspended at the Termination_Attempt DP, and a control relationship has been established.

7.4.3.50.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.50.4 Responding entity (SCF)

7.4.3.50.4.1 Normal procedure

SCF preconditions:

None.

SCF postconditions:

1. An SLPI has been invoked

2. For a TDP-R, an SSF instruction is being prepared.

On receipt of the TerminationAttempt operation, the SCSM moves from "Idle" state to the state "Preparing SSF Instructions". A control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked for processing the TerminationAttempt operation. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked. The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI (i.e., the requested IN service itself).

7.4.3.50.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.51 TerminationNotification procedure

7.4.3.51.1 General description

This operation is sent by the SSF to indicate the outcome of a request for notification when the call exits to the NULL PIC.

7.4.3.51.2 Parameters

– correlationID *(EchoData)*:
  This parameter correlates the SendNotification request with the TerminationNotification response.

– terminationIndicator *(TerminationIndicator)*:
  This parameter is sent to provide a "yes" or "no" value for a particular indicator identified in the ASN.1 information.

– connectTime *(ConnectTime)*:
  This parameter is sent to indicate the elapsed time from answer to disconnect.

– busyCause *(BusyCause)*:
  This parameter is sent to indicate the busy cause.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.51.3 Invoking entity (SSF)

7.4.3.51.3.1 Normal procedure

SSF precondition:

The SSF receives a SendNotification operation.

SSF postcondition:

The SSF sends a TerminationNotification when the call reaches the NULL PIC.

7.4.3.51.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.51.4 Responding entity (SCF)

7.4.3.51.4.1 Normal procedure

SCF precondition:

The SCF sent a SendNotification operation.

SCF postcondition:

The SCSM-FSM returns to state “Idle.”

7.4.3.51.4.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

7.4.3.52 Timeout procedure

7.4.3.52.1 General description

This operation is sent from the SSF to the SCF at the O_Mid_Call DP in the O_Active PIC. After the *TimeoutTimer* has expired, this operation is used to report the timeout event requested by RequestReportBCMEvent. The SCF can respond with an analyzeRoute, continue, forwardCall, releaseCall, or sendToResource operation.

7.4.3.52.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. This parameter identifies the originating facility.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user. Refer to 7.4.6 for population rules for the bearer capability parameter.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

- extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.
- serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:
  If the SSF detects a requested event as an EDP-Notification, then the SSF populates the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of "TRUE."

7.4.3.52.3 Invoking entity (SSF)

7.4.3.52.3.1 Normal procedure

SSF preconditions:

1. The call is answered.
2. The Timeout event is armed.
3. There is an existing control relationship and the EDP *TimeoutTimer* is expired.

SSF postconditions:

1. The SSF-FSM moves to the state "Waiting for Instructions" while the call is still connected.

7.4.3.52.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.52.4 Responding entity (SCF)

7.4.3.52.4.1 Normal procedure

SCF preconditions:

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.

SCF postconditions (EDP):

1. The SCSM-FSM moves to the state "Preparing SSF Instructions".

7.4.3.52.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.53 TMidCall procedure

7.4.3.53.1 General description

This operation is used to indicate that a feature request is received from the terminating party (e.g. hook flash, ISDN feature activation, T1.607 HOLD or RETrieve message). (DP: T_Mid_Call).

7.4.3.53.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  See 3.1. This parameter identifies the originating facility.

– calledPartyNumber *(CalledPartyID)*:
  See T1.113 *Called Party Number* signaling information. This parameter is used to identify the called party in the forward direction.

– callingPartyNumber *(CallingPartyID)*:
  See T1.113 *Called Party Number* signaling information. Refer to 7.4.6 for population rules for this parameter.

– callingPartyBusinessGroupID *(CallingPartyBGID)*:
  See 3.1. The SCF uses this information element to select SLPs based on the group and for authorization purposes.

– calledPartyBusinessGroupID *(CalledPartyBGID)*:
  See 3.1

– callingPartySubaddress *(CallingPartySubaddress)*:
  See T1.113 subaddress information

– calledPartySubaddress *(CallingPartySubaddress)*:
  See T1.113 subaddress information.

– featureRequestIndicator *(FeatureRequestIndicator)*:
  This parameter indicates the type of feature requested.

– carrier *(Carrier)*:
  As defined in 3.1.

7.4.3.53.3 Invoking entity (SSF)

7.4.3.53.3.1 Normal procedure

SSF preconditions:

1. Call origination attempt has been initiated.
2. Indication received from Terminating BCSM that the call is accepted and the terminating party has answered.
3. Feature request received from a terminating party.
4. For a TDP, call gapping or service filtering are not in effect.
5. DP criteria have been met.
6. For a TDP, there is no existing control relationship.
7. For an EDP, there is an existing control relationship and the EDP T_Mid_Call is armed.

SSF postconditions:

1. For a TDP-R, basic call processing has been suspended at T_Mid_Call DP, and a control relationship has been established.
2. For a TDP-N, call processing proceeds to the T_ Active PIC, and no control relationship has been established.

3. For an EDP-R, basic call processing has been suspended at T_Mid_Call DP, and the existing control relationship continues.
4. For an EDP-N, basic call processing proceeds at the T_Active PIC, and the existing relationship continues unless no further EDPs are armed and no “CallInformationReport” or “ApplyChargingReport” is requested.

7.4.3.53.3.2 Error Handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.53.4 Responding entity (SCF)

7.4.3.53.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. the SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” are requested, or
3. the SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.53.4.2 Error Handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.54 TNoAnswer procedure

7.4.3.54.1 General description

This operation is sent from the SSF to the SCF after detecting a valid trigger condition at the T_No_Answer DP, or to report an event requested by RequestReportBCMEvent.

This operation requests the SSF/CCF to send a TNoAnswer TDP-Request operation when it encounters a T_No_Answer trigger.

7.4.3.54.2 Parameters

When the SSF sends this information to the SCF, it should use the same rules as for TerminationAttemptArg, with the following exceptions:

- serviceAddressInformationServiceKey *(UserID)*:
  If the SSF encounters the T_No_Answer trigger, then the SSF shall populate the serviceAddressInformationServiceKey with the identity of the terminating facility or directory number to which the trigger is assigned.
- serviceAddressInformationTriggerType *(TriggerCriteriaType)*:
  The serviceAddressInformationTriggerType parameter shall be set to “tNoAnswer”.
- sap *(Sap)*:
  When the originating access is an SS7 trunk and the received IAM contains the Sap parameter, the SSF shall map the Sap parameter received in the IAM directly to the corresponding Sap parameter in the INAP message.

7.4.3.54.3 Invoking entity (SSF)

7.4.3.54.3.1 Normal procedure

SSF preconditions (TDP):

1. An incoming call has been received.
2. The terminating party has not answered within a specified time period.
3. Call gapping or service filtering are not in effect.
4. DP criteria have been met.
5. For a TDP-R, there is no existing control relationship.

SSF preconditions (EDP):

1. For an EDP-R, there is an existing control relationship and the T_No_Answer EDP is armed.
2. For an EDP-N, there is an existing control or monitoring relationship and the T_No_Answer EDP is armed.

SSF postconditions (TDP):

1. For a TDP-R, basic call processing has been suspended at T_No_Answer DP, and a control relationship has been established.
2. For a TDP-N, default exception handling has been provided, and no control relationship has been established.

SSF postconditions (EDP):

1. The SSF-FSM stays in the state “Monitoring” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested.
2. The SSF-FSM moves to the state “idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” is requested.
3. The SSF-FSM moves to the state “Waiting for Instructions” if the message type was request. Call processing is interrupted.

7.4.3.54.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.54.4 Responding entity (SCF)

7.4.3.54.4.1 Normal procedure

SCF preconditions (TDP):

None.

SCF preconditions (EDP):

1. For an EDP-R at the SSF, an existing control relationship is in place and an SLPI is running.
2. For an EDP-N, an existing monitoring relationship is in place and an SLPI is running.

SCF postconditions (TDP):

1. An SLPI has been invoked.
2. For a TDP-R, a control relationship is established, and an SLPI has been invoked.
3. For a TDP-R, an SSF instruction is being prepared.
4. For a TDP-N, no relationship is established. An SLPI has been invoked, executes and terminates.

SCF postconditions (EDP):

1. For an EDP, the SCSM-FSM stays in the substate “Waiting for Notification or Request” if the message type was notification and there are still EDPs armed or a “CallInformationReport” or “ApplyChargingReport” requested, or
2. the SCSM-FSM moves to the state “Idle” if the message type was notification and there are no more EDPs armed, no “CallInformationReport” or “ApplyChargingReport” are requested, or
3. the SCSM-FSM moves to the state “Preparing SSF Instructions” if the message type was request.

7.4.3.54.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.55 Update procedure

7.4.3.55.1 General description

The Update operation is used by the SCF to request that certain SSF information (e.g., a trigger activation/deactivation, O_No_Answer and T_No_Answer timers) be changed.

7.4.3.55.2 Parameters

– administrableObject *(AdministrableObject)*:
  This parameter contains the TriggerItemAssignment and SSPUserResource parameters.

– amp1 *(Amp1)*:
  This parameter is sent to mark and trace test calls/signals and activate logging on selected calls through the network.

– amp2 *(Amp2)*:
  This parameter contains additional information to mark and trace test calls/signals and activate logging on selected calls through the network.

– extensionParameter *(ExtensionParameter)*:
  This parameter is sent to provide network-specific information.

7.4.3.55.3 Invoking Entity (SCF)

7.4.3.55.3.1 Normal Procedure

The Update operation performs an update of new triggers and any associated no answer timers, as well as the ability to update the status of a Message Waiting Indicator (MWI) in the SSP. The Update message allows the SCP to update attributes associated with Termination Attempt triggers. The SCP can control several data parameters in the SSP, which would affect such attributes as the service-provider ID, trigger activation state, MWI, etc. The SSP responds with an Update Success message that gives the result of the request.

7.4.3.55.3.2 Error Handling

Generic error handling of the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.55.4 Responding Entity (SSF)

7.4.3.55.4.1 Normal Procedure

The SSP shall expect the Update message to include the local parameter or field to be updated. The SSP shall update the value of the information based on the parameter value received from the SCP. Some examples of the Update Message parameters, as contained as part of the AdministerableObject, include the following:

– ActivationStateCode

– Mwi

– *ONoAnswerTimer*

– *TNoAnswerTimer*

7.4.3.55.4.2 Error Handling

Generic error handling of the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.56 UpdateRequest procedure

7.4.3.56.1 General description

The UpdateRequest operation is used by the SCF to request that certain SSF information (e.g., a trigger activation/deactivation) be changed. Note that the Update operation is a successor to the UpdateRequest operation.

7.4.3.56.2 Parameters

– serviceAddressInformationServiceKey *(UserID)*:
  This parameter is sent to provide an identifier of a user.

– triggerCriteriaFlag *(TriggerCriteriaFlag)*:
  This parameter is sent to indicate whether a trigger should be active or inactive.

– bearerCapability *(BearerCapability)*:
  This parameter indicates the type of bearer capability connection to the user.

– tNoAnswerTimer *(TNoAnswerTimer)*:
  This parameter is sent to indicate the value, in seconds, of the SSF terminating no answer timer.

7.4.3.56.3 Invoking Entity (SCF)

7.4.3.56.3.1 Normal Procedure

The SCF shall populate the relevant parameters and send this operation based on the service logic. The SCF may send the Update_Request message to update trigger activation status in the SSP. The SSP responds with an Update_Data message that gives the result of the request.

The SCP may also send an Update_Request in a TCAP Query Package to the SSP. In this case the SSP shall send a Response Package message with the Return Result component to tell the SCP that the request was honored. The Return Result shall contain no parameters. Failure messages shall contain the Failure Cause parameter coded to either UnavailableResources, or RateTooHigh.

The Update_Request message may also be used to request an SSP to activate and deactivate an Off-Hook Immediate, Off-Hook Delay, or a Termination Attempt trigger assigned to a particular facility. The trigger must have previously assigned in the SSP.

7.4.3.56.3.2 Error Handling

Generic error handling of the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.3.56.4 Responding Entity (SSF)

7.4.3.56.4.1 Normal Procedure

The SSF shall update the local value of the information based on the parameter values received from the SCP. Procedures differ based on the manner in which the Update_Request is sent by the SCP:

- If the Update_Request is received in a TCAP Conversation Package, the SSP shall send the Conversation Package with Return Result to tell the SCP the request has been honored.
- If the Update_Request is sent in a TCAP query Package, the SSP, after completing the request, shall send a Response Package with Return Result to the SCP.
- If the request cannot be honored, the SSP shall return an Update_Data Return Result to the SCP.
- If the SSP experiences an outage or fault, and goes through a restart, the triggers shall be restored to the last known state.

7.4.3.56.4.2 Error Handling

Generic error handling of the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.4 Detailed operation procedures: SCF-SRF interface

7.4.4.1 AssistRequestInstructions procedure

7.4.4.1.1 General description

The AssistRequestInstructions operation is used in the direct SCF-SRF information transfer case. It is sent from the SRF directly to the SCF for instructions when the StrParameterBlock contains a CorrelationBlock.

7.4.4.1.2 Parameters

- sCFCorrelationID *(EchoData)*
  The *sCFCorrelationID* parameter is used by the SCF to correlate the messages from the SRF.
- extensionParameter *(ExtensionParameter)*:
  This parameter indicates an extension of an argument data type. Its content is network-specific.

7.4.4.1.3 Invoking entity (SRF)

7.4.4.1.3.1 Normal procedure

SRF preconditions:

1. A control relationship exists between the SCF and SSF
2. SendToResource operation has been received by the SSF.
3. The bearer connection has been established between the SSF and the SRF.
4. The correlationBlock sent from the SSF has been received by the SRF.

SRF postcondition:

1. The SRF shall start a timer and wait for a response to be returned from the SCF

7.4.4.1.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.4.1.4 Responding entity (SCF)

7.4.4.1.4.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and the SSF
2. SendToResource operation has been sent to the SSF with the correlationBlock.
3. The SLPI intends to provide instructions to the SRF directly.

SCF postconditions:

1. A direct control relationship is established between the SCF and the SRF
2. The SCF shall populate the parameters and send an InstructionsToSRF operation based on Service Logic.

Once the AssistRequestInstructions operation is received by the SCF, it then responds through the direct connection with an InstructionsToSRF message to instruct the SRF to perform certain functions, such as play announcement and/or collect digits. If additional information needs to be passed between the SCF and the SRF, the CallInfoFromResource and CallInfoFromResourceRR operations are used.

7.4.4.1.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.4.2 CallInfoFromResource procedure

See 7.4.3.9.

7.4.4.3 CallInfoFromResourceRR procedure

See 7.4.3.10.

7.4.4.4 CancelResourceEvent procedure

See 7.4.3.13. Although this operation is between the SCF and the SSF, it impacts the SCF to SRF connection.

7.4.4.5 InstructionsToSRF procedure

7.4.4.5.1 General description

The InstructionsToSRF operation is used in the direct SCF-SRF information transfer case. It is a response to the AssistRequestInstructions from the SRF. The parameters of the InstructionsToSRFArg specify the particular function(s) that the SRF should perform.

7.4.4.5.2 Parameters

– resourceType *(ResourceType)*:
The *resourceType* parameter identifies the specific resource type (or capability) that needs to be provided to the user.

– strParameterBlock *(StrParameterBlock)*:
The *strParameterBlock* parameter is sent to provide any information needed for the capability specified by the *resourceType* parameter. Therefore, it can take one of a number of different codings depending on the capability that is to be utilized. Each resource type value implies its own parameter block and its own coding. Examples include “Play Announcement” and “Play Announcement and Collect Digits.” Information collected from a user is returned to the SCF in a CallInfoFromResource operation directly.

– disconnectFromIPForbidden *(DisconnectFromIPForbidden):*
This parameter is used to allow the SRF initiated disconnect on the bearer connection if it is set to FALSE.

– extensionParameter *(ExtensionParameter)*:
This parameter is sent to provide network-specific information.

7.4.4.5.3 Invoking entity (SCF)

7.4.4.5.3.1 Normal procedure

SCF preconditions:

1. A control relationship exists between the SCF and SSF
2. AssistRequestInstructions operation has been received by the SCF
3. The SLPI intends to provide instructions to the SRF
4. The SCF shall populate the parameters and send this operation based on Service Logic.

SCF postcondition:

1. The SCF shall start a timer and wait for a response to be returned from the SRF

7.4.4.5.3.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.4.5.4 Responding entity (SRF)

7.4.4.5.4.1 Normal procedure

SRF preconditions:

1. A control relationship exists between the SCF and the SSF
2. AssistRequestInstructions operation has been sent to the SCF

SRF postconditions:

1. A control relationship exists between the SCF and the SRF
2. The SRF shall execute the instructions received and report the results back to the SCF directly.

Once the InstructionsToSRF operation is received by the SRF, it begins to perform the requested functions, such as play announcement and/or collect digits. If additional information needs to be passed between the SCF and the SRF, the CallInfoFromResource and CallInfoFromResourceRRoperations are used. Within the CallInfoFromResourceRR operation and the InstructionsToSRF operation, there is a disconnectFromIPForbidden parameter. This parameter is used to allow the SRF initiated disconnect if it is set to FALSE (the default value is TRUE). The final result is reported back to the SCF directly in a CallInfoFromResource operation. When the SCF receives the final result, it sends a CallInfoFromResourceRR operation to the SRF and set disconnectFromIPForbidden parameter to FALSE. The SRF will then disconnect the bearer connection with the SSF. Alternatively, the SCF can send a CancelResourceEvent message to the SSF to disconnect the bearer channel to the SRF. The SCF can send the CancelResourceEvent message regardless of what the value is set in the disconnectFromIPForbidden parameter to the SRF.

7.4.4.5.4.2 Error handling

Generic error handling for the operation related errors is described in 7.4.2 and the TCAP services that are used for reporting operation errors are described in 7.4.7.

7.4.4.6 ResourceClear procedure

See 7.4.3.41. Although this operation is between the SCF and the SSF, it impacts the SCF to SRF connection.

7.4.4.7 SendToResource procedure

See 7.4.3.45. Although this operation is between the SCF and the SSF, it impacts the SCF to SRF connection.

7.4.5 Detailed operation procedures: SCF-SDF interface

For background information, refer to informative Annex C.

7.4.6 Population rules for SSF/CCF to SCF Operations

T1 IN parameter population rules are contained in this clause for the operations listed in Table 15. The parameters are declared in the sequence order in which they are defined in this standard. Only those parameters that are impacted by the PIC preceding the DP are reported for EDPs.

Table 15 - Operation parameters

Clause	Operation	Parameter
7.4.6.1	Analyzed Information	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: calledPartyNumber *(CalledPartyID)*: servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: chargeNumber *(ChargeNumber)*: callingPartyNumber *(CallingPartyID)*: callingPartyBusinessGroupID *(CallingPartyBGID)*: callingPartysCategory *(ChargePartyStationType)*: carrier *(Carrier)*: accessCode *(AccessCode)*: dialedDigitsCAI *(CollectedAddressInfo)*: dialedDigitsCD *(CollectedDigits)*: featureCode *(VerticalServiceCode)*: travelingClassMark (*Tcm)*: originalCalledPartyID *(OriginalCalledPartyID)*: redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sap *(Sap)* sTRConnection (*STRConnection)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)* genericAddressList *(GenericAddressList)* networkSpecificFacilities *(NetworkSpecificFacilities)* forwardCallIndicator *(ForwardCallIndicator)* genericDigitsList *(GenericDigitsList)* callingGeodeticLocation *(CallingGeodeticLocation)*

Clause	Operation	Parameter
7.4.6.2	Collected Information	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: chargeNumber *(ChargeNumber)*: servingAreaID *(Lata)*: carrier *(Carrier)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: accessCode *(AccessCode)*: dialedDigitsCAI *(CollectedAddressInfo)*: dialedDigitsCD *(CollectedDigits)*: featureCode (*VerticalServiceCode)*: travelingClassMark (*Tcm)*: originalCalledPartyID *(OriginalCalledPartyID)*: redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sap *(Sap)* genericAddressList *(GenericAddressList)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)* forwardCallIndicator *(ForwardCallIndicator)* genericDigitsList *(GenericDigitsList)* callingGeodeticLocation *(CallingGeodeticLocation)*

Clause	Operation	Parameter
7.4.6.3	Facility Selected and Available	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: amp1 *(Amp1)* amp2 *(Amp2)* extensionParameter *(ExtensionParameter)* calledPartyNumber *(CalledPartyID)* servingAreaID *(Lata)* serviceAddressInformationTriggerType *(TriggerCriteriaType)* chargeNumber *(ChargeNumber)* callingPartyNumber *(CallingPartyID)* callingPartysCategory *(ChargePartyStationType)* originalCalledPartyID *(OriginalCalledPartyID)* redirectingPartyID *(RedirectingPartyID)* redirectionInformation *(RedirectionInformation)* calledPartysCategory *(CalledPartyStationType)* sap *(Sap)* displayInformationGN *(GenericName)* cGEncountered *(ACGEncountered)* sTRConnection (*STRConnection)* aMASequenceNumber *(AMASequenceNumber)*
7.4.6.4	O Answer	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: amp1 *(Amp1)* amp2 *(Amp2)* extensionParameter *(ExtensionParameter)* localSSPTime *(LocalSSPTime)*

Clause	Operation	Parameter
7.4.6.5	O Called Party Busy	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: calledPartyNumber *(CalledPartyID)*: servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: chargeNumber *(ChargeNumber)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: carrier *(Carrier)*: originalCalledPartyID *(OriginalCalledPartyID*): redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: busyCause *(BusyCause)*: sap (*Sap)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sTRConnection (*STRConnection)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)*
7.4.6.6	O Disconnect	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: amp1 *(Amp1)* amp2 *(Amp2)* extensionParameter *(ExtensionParameter)* disconnectCause *(DisconnectCause)* localSSPTime *(LocalSSPTime)*
7.4.6.7	O DTMF Entered	serviceAddressInformationServiceKey *(UserID)* bearerCapability *(BearerCapability)* serviceAddressInformationMiscCallInfo *(NotificationIndicator)* dTMFDigitsDetected *(DTMFDigitsDetected)* amp1 *(Amp1)* amp2 *(Amp2)* extensionParameter *(ExtensionParameter)* localSSPTime *(LocalSSPTime)*

Clause	Operation	Parameter
7.4.6.8	O No Answer	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: calledPartyNumber *(CalledPartyID)*: servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: chargeNumber *(ChargeNumber)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: carrier *(Carrier)*: originalCalledPartyID *(OriginalCalledPartyID)*: redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: sap *(Sap)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sTRConnection (*STRConnection)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)*
7.4.6.9	Origination Attempt	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: chargeNumber *(ChargeNumber*): servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: carrier *(Carrier)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sap *(Sap)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)* serviceAddressInformationMiscCallInfo *(NotificationIndicator)*

Clause	Operation	Parameter
7.4.6.10	O Suspended	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)* amp1 *(Amp1)* amp2 *(Amp2)* extensionParameter *(ExtensionParameter)*
7.4.6.11	O Term Seized	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: amp1 *(Amp1)* amp2 *(Amp2)* extensionParameter *(ExtensionParameter)*
7.4.6.12	Route Select Failure	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: chargeNumber *(ChargeNumber)*: servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: calledPartyNumber *(CalledPartyID)* travelingClassMark (*Tcm)*: originalCalledPartyID *(OriginalCalledPartyID)*: carrier *(Carrier)*: redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sap *(Sap)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)* serviceAddressInformationMiscCallInfo *(NotificationIndicator)*
7.4.6.13	T Answer	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: amp1 *(Amp1)* amp2 *(Amp2)* extensionParameter *(ExtensionParameter)*

Clause	Operation	Parameter
7.4.6.14	T Busy	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability):* calledPartyNumber *(CalledPartyID)*: servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: chargeNumber *(ChargeNumber)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: originalCalledPartyID *(OriginalCalledPartyID)*: redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: busyCause (BusyCause): busyType *(BusyType)*: calledPartysCategory *(CalledPartyStationType*): sap *(Sap)*: displayInformationGN *(GenericName)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sap *(Sap)* sTRConnection (*STRConnection)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)*

Clause	Operation	Parameter
7.4.6.15	Termination Attempt	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: calledPartyNumber *(CalledPartyID)*: servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: calledPartysCategory *(CalledPartyStationType)*: chargeNumber *(ChargeNumber)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: travelingClassMark *(Tcm)*: originalCalledPartyID *(OriginalCalledPartyID)*: redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: displayInformationGN *(GenericName)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sap *(Sap)* sTRConnection *(STRConnection)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)* genericDigitsList *(GenericDigitsList)* callingGeodeticLocation *(CallingGeodeticLocation)*
7.4.6.16	Timeout	amp1 *(Amp1)* amp2 *(Amp2)* bearerCapability *(BearerCapability)* extensionParameter *(ExtensionParameter)* serviceAddressInformationMiscCallInfo *(NotificationIndicator)* serviceAddressInformationServiceKey *(UserID)*

Clause	Operation	Parameter
7.4.6.17	T No Answer	serviceAddressInformationServiceKey *(UserID)*: bearerCapability *(BearerCapability)*: calledPartyNumber *(CalledPartyID)*: servingAreaID *(Lata)*: serviceAddressInformationTriggerType *(TriggerCriteriaType)*: chargeNumber *(ChargeNumber)*: callingPartyNumber *(CallingPartyID)*: callingPartysCategory *(ChargePartyStationType)*: originalCalledPartyID *(OriginalCalledPartyID)*: redirectingPartyID *(RedirectingPartyID)*: redirectionInformation *(RedirectionInformation)*: calledPartysCategory *(CalledPartyStationType)*: sap *(Sap)*: displayInformationGN *(GenericName)*: serviceAddressInformationMiscCallInfo *(NotificationIndicator)*: cGEncountered *(ACGEncountered)*: amp1 *(Amp1)* amp2 *(Amp2)* sap *(Sap)* sTRConnection (*STRConnection)* aMASequenceNumber *(AMASequenceNumber)* extensionParameter *(ExtensionParameter)*

7.4.6.1 Analyzed Information

When the SSF sends this information to the SCF, the following rules apply:

serviceAddressInformationServiceKey *(UserID)*:

The serviceAddressInformationServiceKey *(UserID)* parameter shall contain the identity of the originating facility. The *UserID* may be a DN (e.g., for non-ISDN interfaces), ISDNI, TrunkGroupID (for public trunks), or PrivateFacilityGID (for private trunks). A DN included in this parameter shall be the calling party number of the facility, not necessarily its billing number. If a call previously encountered call forwarding, the serviceAddressInformationServiceKey *(UserID)* is the identity of the last forwarding station.

bearerCapability *(BearerCapability)*:

The bearerCapability *(BearerCapability)* parameter shall contain the bearer capability of the call that encountered the trigger. The SSF shall include the bearerCapability *(BearerCapability)* parameter in the message regardless of the originating access type.

If the originating access is a non-ISDN line dedicated to PSDS calls, the bearerCapability *(BearerCapability)* parameter shall indicate "56 kbps Circuit Mode Data."

If a non-ISDN line is not dedicated to PSDS calls, the bearerCapability *(BearerCapability)* parameter shall indicate “3.1 kHz.”

If the originating access is an ISDN interface the bearerCapability *(BearerCapability)* parameter shall be mapped from the information received.

If the originating access is a conventional trunk or private facility dedicated to PSDS calls, the bearerCapability *(BearerCapability)* parameter shall indicate “56 kbps Circuit Mode Data.”

If the conventional trunk or private facility is not dedicated to PSDS calls, the bearerCapability *(BearerCapability)* parameter shall indicate “3.1 kHz.”

If the originating access is an SS7 trunk, the bearerCapability *(BearerCapability)* parameter shall be mapped from the information transfer capability sub-field of the User Service Information parameter received in the IAM.

calledPartyNumber *(CalledPartyID)*:

The calledPartyNumber *(CalledPartyID)* parameter shall contain the address of the called party. This parameter shall be included when the SSF encounters a Specific_Digit_String or N11 (e.g., Emergency Services) trigger. This calledPartyNumber *(CalledPartyID)* shall contain a 10-digit NANP number, an international number, a network-specific number, or an N11 code.

When the originating access is a non-ISDN line or an ISDN interface using keypad, the SSF shall encode the calledPartyNumber *(CalledPartyID)* parameter using the results of digit analysis.

When the originating access is an ISDN interface that sent a *Called Party Number* information element, the information received in the *Called Party Number* information element shall be mapped to the calledPartyNumber *(CalledPartyID)* parameter.

When the originating access is a MF trunk using either conventional or Equal Access signaling, the SSF shall encode the calledPartyNumber *(CalledPartyID)* parameter as follows:

Nature of Number - The nature of number field shall be coded “National.”

Numbering Plan - The numbering plan shall be coded “ISDN Numbering Plan.”

Digits - The digits field shall be coded with the digits received.

When an NANP address is received, the digits field of the calledPartyNumber *(CalledPartyID)* shall be populated with a 10-digit address derived from the digits received.

When the originating access is a private facility trunk, the SSF shall encode the *CalledPartyID* parameter using the same rules as defined for a non-ISDN line. calledPartyNumber *(CalledPartyID)*

When the originating access is an SS7 trunk, the *Called Party Number* parameter received in the IAM shall be mapped to the calledPartyNumber *(CalledPartyID)* parameter.

servingAreaID *(Lata)*:

The servingAreaID *(Lata)* parameter shall contain the LATA of the originating facility.

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set as indicated in Table 16.

Table 16 - Analyzed Information: Trigger Criteria Type

Trigger Encountered	**serviceAddressInformationTriggerType** *(TriggerCriteriaType)*
BRI_Feature_Activation_Indicator	featureActivator
Public_Feature_Code	verticalServiceCode
Specific_Feature_Code	specificFeatureCode
Customized_Dialing_Plan	customizedAccess, customizedIntercom
Specific_Digit_String	npa, npaN, npaNX, npaNXX, npaNXXX, npaNXXXX, npaNXXXXX, npaNXXXXXX
N11	n11

chargeNumber *(ChargeNumber)*:

The chargeNumber *(ChargeNumber)* parameter shall contain the ANI of the originating facility, or the ANI received in MF signaling, or the charge number received in ISUP signaling. This value will be a 10-digit NANP number. If a call previously encountered call forwarding, the *ChargeNumber* is the ANI of the last forwarding station.

If the originating access is a non-ISDN line or an ISDN interface, then the SSF shall include the ChargeNumber information for the non-ISDN line or ISDN interface.

If the originating access is an SS7 trunk and the received IAM contains the charge number parameter or the calling party number parameter, then the SSF shall include the chargeNumber *(ChargeNumber)* parameter. If the charge number parameter is included in the IAM, the SSF shall use this parameter to formulate the chargeNumber *(ChargeNumber)* parameter. If the charge number is not included in the IAM and the calling party number and OLI parameters[3] are included, the SSF shall use the calling party number parameter to formulate the chargeNumber *(ChargeNumber)* parameter. If neither *ChargeNumber* nor *CallingPartyNumber* are included in the IAM, the SSF shall have the ability to use an NPA-NXX code associated with the incoming trunk group as the ANI information sent to the SCF in the query message.

If the originating access is an MF trunk and the ANI is received in MF signaling, then the SSF shall include the chargeNumber *(ChargeNumber)* parameter. In the case of Equal Access Signaling, the ANI received shall be mapped to the chargeNumber *(ChargeNumber)* parameter. When only 3 digits of ANI are received, the SSF shall have the ability to use a 3-digit office code associated with the incoming trunk group to construct 6 digits of ANI, which are sent to the SCF. In the case of Centralized Automatic Message Accounting (CAMA) signaling, the SSF shall append the appropriate NPA code to the 7 digits of ANI received, and populate the chargeNumber *(ChargeNumber)* parameter with the resulting 10 digits of ANI. When no ANI information is received, the SSF shall have the ability to use a 3-digit office code and an NPA code associated with the incoming trunk group to construct 6 digits of ANI, which are sent to the SCF.

[3] These are all the OLI values support by IN (a subset of the OLI values, that are the binary equivalent of the II digits administered by North American Numbering Plan Administration, http://www.nanpa.com). In addition, IN uses one of the reserved values, defining value (64) as "IN."

If the originating access is an MF trunk, and the signaling method used does not support the transport of ANI information (e.g., Local Automatic Message Accounting (LAMA_ signaling is used), the SSF shall include the chargeNumber *(ChargeNumber)* parameter, and this parameter shall be populated with a 6-digit number (NPA-NXX) that is provisioned against this trunk group specifically for this purpose.

callingPartyNumber *(CallingPartyID)*:

The callingPartyNumber *(CallingPartyID)* parameter contains the DN of the originating line or the calling party number received in ISUP signaling. This value may be a 10-digit NANP number or an international number. If a call previously encountered call forwarding, the callingPartyNumber *(CallingPartyID)* is the calling number of the original caller, not the forwarding station.

For an ISDN interface, the callingPartyNumber *(CallingPartyID)* shall be the user-provided number (if provided and passed screening). Otherwise, the callingPartyNumber *(CallingPartyID)* shall be the network-provided number.

If the originating access is an SS7 trunk and this information is available in the IAM, then the SSF shall include the callingPartyNumber *(CallingPartyID)* parameter.

If the originating access is a non-ISDN line or an ISDN interface, then the SSF shall include the callingPartyNumber *(CallingPartyID)* parameter.

If the originating access is a trunk using MF signaling, then the SSF shall not include the callingPartyNumber *(CallingPartyID)* parameter.

callingPartyBusinessGroupID *(CallingPartyBGID)*:

The SSF shall include the callingPartyBusinessGroupID *(CallingPartyBGID)* parameter in the AnalyzedInformationArg when the following conditions exist:

> The originating access (non-ISDN line, ISDN interface, or private-facility trunk group) is assigned to a Basic Business Group (BBG) or Multi-Switch Business Group (MBG). The originating access in this case may be associated with the caller or a user forwarding the call via a switch-based call forwarding feature.
>
> The AnalyzedInformationArg is sent in response to a CDP trigger.
>
> The Business Group parameter is received in an ISUP IAM and the AnalyzedInformationArg is sent in response to a Specific_Digit_String trigger.

A callingPartyBusinessGroupID *(CallingPartyBGID)* parameter shall contain information associated with the calling party. If the call has been forwarded, the parameter shall contain information associated with the last party that forwarded the call.

The SSF shall retain the business group information for subsequent call processing. If the call is to be routed over an SS7 trunk, the SSF shall include the information in the Business Group parameter of the ISUP IAM.

callingPartysCategory *(ChargePartyStationType)*:

The callingPartysCategory *(ChargePartyStationType)* contains the OLI of the originating line, the OLI received in ISUP signaling, or the II digits received in MF signaling[3]. If the call previously encountered call forwarding, the *ChargePartyStationType* is the OLI or II digits of the last forwarding station.

When the originating access is an Equal Access MF trunk, the II digits shall be mapped to the appropriate value of ChargePartyStationType. The binary equivalent of the decimal II digits shall be used as the value for ChargePartyStationType (e.g., II digits of “06” are equivalent to a value of “00000110”).

When the originating access is an SS7 trunk and the received IAM contains an OLI parameter, the SSF shall include the callingPartysCategory *(ChargePartyStationType)* parameter. The value of the OLI parameter shall be directly mapped to the callingPartysCategory *(ChargePartyStationType)* parameter. When the OLI parameter is not included, the ANI II digits pair “23” shall be assigned to the call and sent to the database in the query.

When the originating access is a non-ISDN line or an ISDN interface, the callingPartysCategory *(ChargePartyStationType)* parameter shall be coded using the Class of Service information that shall be mapped to the appropriate value. This is determined according to the rules used to determine the MF II digits.

carrier *(Carrier)*:

The carrier *(Carrier)* parameter consists of two fields: Carrier Selection and Carrier ID.

The Carrier Selection field indicates whether the primary carrier was presubscribed or dialed. This field shall be populated as follows:

> When the originating access is a non-ISDN line or an ISDN interface, the SSF shall use the subscriber’s carrier presubscription information along with the dialed digits to encode the Carrier Selection field.
>
> When the originating access is a conventional trunk, the SSF shall map any received presubscription indication into the Carrier Selection field. (This indication comes from the KP or KP´ signal.).
>
> When the originating access is an SS7 trunk, the SSF shall map the information received in the carrier selection parameter to the Carrier Selection field.

If the SSF encounters a Public_Feature_Code trigger or a Customized_Dialing_Plan trigger, and the caller dials a Carrier Access Code (i.e., 101XXXX), then the SSF shall include the CIC (i.e., XXXX) in the Carrier ID field of the carrier *(Carrier)* parameter, with the proper coding for the Carrier Selection field. If the caller does not dial a Carrier Access Code, then the Carrier ID field shall contain the presubscribed carrier for an InterLATA call.

If the SSF encounters a Specific_Digit_String trigger or an N11 trigger, then the Carrier ID field shall contain the carrier to be used to route the call. This carrier may be determined by the analysis in the Analyze_Information PIC, specified by a dialed 101XXXX code, or specified in EAMF or ISUP signaling. The Carrier ID field shall contain four digits. If the SSF receives a CIC containing three digits (XXX), then the Carrier ID field shall be coded 0XXX.

When the call is received from an SS7-supported trunk and the IAM contains the Carrier Identification Parameter (CIP), the SSF shall include the carrier (Carrier) parameter.

accessCode *(AccessCode)*:

The accessCode *(AccessCode)* parameter is included for CDP Access Code triggers only, and shall contain the access code that caused the trigger. Access codes that are not IN triggers shall not be sent to the SCF, but shall be processed by the switch.

dialedDigitsCAI *(CollectedAddressInfo)*:

If the SSF encountered a CDP Intercom Code trigger then the SSF shall use these digits to populate the dialedDigitsCAI *(CollectedAddressInfo)* parameter.

When the originating access is a non-ISDN line or ISDN interface using keypad (including the case when the *Called Party Number* information element is sent with the nature of number and numbering plan field set to "unknown") with a private numbering plan, or a private facility, the SSF shall encode the dialedDigitsCAI *(CollectedAddressInfo)* parameter as follows:

Nature of Number - The Nature of Number field shall be coded "not applicable."

Numbering Plan - The numbering plan shall be coded "Private."

Digits - The digits field shall be coded with the digits received.

This parameter is also used to carry subsequent digits collected after the CDP Access Code, BRI_Feature_Activation_Indicator, or Public_Feature_Code triggers (when collected according to the normal dialing plan).

dialedDigitsCD (CollectedDigits):

If the SSF collected a variable number of digits using normal interdigit timing and an end of dialing indicator, then the SSF shall use these digits to populate the dialedDigitsCD (*CollectedDigits*) parameter as follows:

Nature of Number - The nature of number field shall be coded "not applicable."

Numbering Plan - The numbering plan field shall be set to "unknown or not applicable."

Digits - The Digits field shall contain the collected digits.

featureCode (VerticalServiceCode):

The featureCode (VerticalServiceCode) parameter is included only for Public Office Dialing Plan Feature Code triggers, and shall contain the value of the vertical service code that caused the trigger. The ∗ or 11 digits are included, if dialed.

travelingClassMark (Tcm):

The travelingClassMark (Tcm) parameter contains the traveling class mark of the calling party.

When the originating access is an SS7-supported trunk, the SSF shall determine if the IAM received contains a Remote Operations (ROS) parameter with a TCM operation. If so, the SSF shall populate the travelingClassMark (Tcm) parameter with the digits taken from the ROS parameter. If the originating access is a private facility, the travelingClassMark (Tcm) parameter shall be coded with the digits received.

The SSF shall include the travelingClassMark (Tcm) parameter when the originating access is a PRI and the SETUP message contains a Facility information element with a TCM operation. The SSF shall populate the travelingClassMark (Tcm) parameter with the TCM digits from the Facility information element.

originalCalledPartyID *(OriginalCalledPartyID)*:

The originalCalledPartyID *(OriginalCalledPartyID)* is the address of the first party that redirected the call. This value shall be a 10-digit NANP number.

When the originating access is a non-ISDN line or an ISDN interface, the SSF shall include the originalCalledPartyID *(OriginalCalledPartyID)* parameter, if this information is available. This information would only be available when the call has been forwarded at least once.

When the originating access is an SS7 trunk and the received IAM contains the original called number parameter, the SSF shall use this information to populate the *OriginalCalledPartyID* parameter.

redirectingPartyID *(RedirectingPartyID)*:

The redirectingPartyID *(RedirectingPartyID)* is the address of the last party that redirected the call. This value shall be a 10-digit NANP number.

When the originating access is a non-ISDN line or an ISDN interface, the SSF shall include the *redirectingPartyID (RedirectingPartyID)* parameter, if this information is available. This information would only be available when the call has been forwarded at least twice.

When the originating access is an SS7 trunk and the received IAM contains the redirecting number parameter, the SSF shall use this information to populate the *redirectingPartyID (RedirectingPartyID)* parameter.

redirectionInformation *(RedirectionInformation)*:

The redirectionInformation *(RedirectionInformation)* parameter contains the original redirecting reason (i.e., the reason for the first redirection of the call), the redirecting reason (i.e., the reason for the last redirection of the call), and the ISUP redirection counter.

When the originating access is a non-ISDN line or an ISDN interface, the SSF shall include the *redirectionInformation* (*RedirectionInformation)* parameter, if this information is available.

When the originating access is an SS7 trunk and the received IAM contains the redirection information parameter, the SSF shall use this information to populate the *redirectionInformation (RedirectionInformation)* parameter.

cGEncountered *(ACGEncountered)*:

When an ACG control is encountered by the SSF and the gap timer has expired or the control has a gap interval of zero, the SSF shall include an *cGEncountered (ACGEncountered)* parameter in the Query Package, indicating that there is an ACG control in effect on this GTA/TT and the number of digits under control.

GenericDigitsList *(GenericDigitsList)*:

This parameter includes a GenericDigits parameter that contains signaled Location Information for a wireline Emergency Caller or cell site and sector information for a wireless Emergency Caller (See T1.628 for Emergency Calling Service description and usage).

CallingGeodeticLocation *(CallingGeodeticLocation)*:

This parameter contains the latitude and longitude information of an Emergency Caller (See T1.628 for Emergency Calling Service description and usage).

7.4.6.2 Collected Information

When the SSF sends this information to the SCF, it should use the same rules as for AnalyzedInformation with the following exceptions:

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType parameter shall be set as indicated in Table 17.

Table 17 - Collected Information: Trigger Criteria Type

Trigger Encountered	serviceAddressInformationTriggerType
Off-Hook Delay	offHookDelay
Channel Setup PRI	channelSetupPRI
Shared Interoffice Trunk	sharedIOTrunk

carrier *(Carrier)*:

The carrier *(Carrier)* parameter consists of two fields: Carrier Selection and Carrier ID. These fields are populated as follows:

> If the caller dials a Carrier Access Code (i.e., 101XXXX), then the SSF shall include the CIC (i.e., XXXX) in the Carrier ID field of the carrier *(Carrier)* parameter, with the proper coding for the Carrier Selection field. If the caller does not dial a Carrier Access Code, then the Carrier ID field shall contain the presubscribed carrier.

7.4.6.3 Facility Selected and Available

When the SSF sends this information to the SCF, it should use the same rules as for OAnswer.

7.4.6.4 O Answer

When the SSF sends this information to the SCF, it should use the following rules:

serviceAddressInformationServiceKey *(UserID)*:

If the SSF detects a requested event as an EDP-Notification, then the SSF shall send an EDP-Notification message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction.

bearerCapability *(BearerCapability)*:

When the SSF sends bearerCapability *(BearerCapability)* to the SCF, it should use the same rules as for AnalyzedInformation.

serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:

If the SSF detects a requested event as an EDP-Notification, then the SSF populates the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of “TRUE.”

localSSPTime *(LocalSSPTime):*

This parameter is sent to provide SCF a time stamp so that the SCF can calculate the call duration.

7.4.6.5 O Called Party Busy

When the SSF sends this information to the SCF, it should use the same rules as for AnalyzedInformation, with the following exceptions:

serviceAddressInformationServiceKey *(UserID)*:

If the SSF detects a TDP-Request, the SSF shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the identity of the originating facility or DN to which the trigger type is assigned. If a call previously encountered call forwarding, the serviceAddressInformationServiceKey *(UserID)* is the identity of the last forwarding station.

If the SSF detects a requested event as an EDP-Request, then the SSF shall send an EDP-Request message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to "oCalledPartyBusy".

busyCause *(BusyCause)*:

If the terminating access is a non-ISDN line, or an ISDN interface for which network-determined user-busy has been detected, the SSF shall populate the busyCause *(BusyCause)* parameter as indicated in Table 18. The SSF detects network-determined user-busy when the originating call portion has been notified of the detection of the T_Busy event specifying user busy.

Table 18 - O Called Party Busy: BusyCause from ISDN/non-ISDN access

busyCause *(BusyCause)* fields	value
Coding Standard	"CCITT-standard"
General Location	"public network serving the remote user"
Extension Bit	"diagnostic is not included after octet 2"
Cause Value Class	"normal event"
Cause	"user busy"

If the terminating access is an ISDN interface and user-determined user-busy has been detected, the SSF shall map the received cause information element directly to the busyCause *(BusyCause)* parameter. The SSF detects user-determined user-busy when the originating call portion has been notified of the detection of the CallRejected event specifying user busy.

If the terminating access is an SS7 trunk, the SSF shall map the received cause indicator parameter in the REL message directly to the busyCause *(BusyCause)* parameter.

The SSF shall populate the Cause field of the busyCause *(BusyCause)* parameter with the values indicated in table19:

Table 19 - O Called Party Busy: BusyCause from SS7 access

Event	Cause value
An analog line being out of order	"Destination out of order"
An ISDN interface being out of order	"Destination out of order"
Maintenance actions	"Destination out of order"
DN not assigned to office equipment	"No route to destination"
Termination Denied	"Call rejected"
Call rejected event not specifying user busy	"Call rejected"

serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:

If the SSF detects a requested event as a Request, then the SSF shall send a EDP-Request message, and shall populate the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of "FALSE."

sap *(Sap)*:

When the originating access is an SS7 trunk and the received IAM contains the *Sap* parameter, the SSF shall map the *Sap* parameter received in the IAM directly to the corresponding *Sap* parameter in the INAP message.

7.4.6.6 O Disconnect

When the SSF sends this information to the SCF, it should use the same rules as for O Answer.

7.4.6.7 O DTMF Entered

When the SSF sends this information to the SCF, it should use the same rules as for O No Answer, with the following exceptions:

dTMFDigitsDetected (DTMFDigitsDetected):

This parameter is sent to indicate the actual DTMF digits collected.

localSSPTime *(LocalSSPTime):*

This parameter is sent to provide SCF a time stamp so that the SCF can calculate the call duration.

7.4.6.8 O No Answer

When the SSF sends this information to the SCF, it should use the same rules as for AnalyzedInformation, with the following exceptions:

serviceAddressInformationServiceKey *(UserID)*:

If the SSF detects a TDP-Request, the SSF shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the identity of the originating facility or DN to which the trigger type is assigned. If a call previously encountered call forwarding, the serviceAddressInformationServiceKey *(UserID)* is the identity of the last forwarding station.

If the SSF detects a requested event as an EDP-Notification, then the SSF shall send an EDP-Notification message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction.

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to “oNoAnswer”.

serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:

If the SSF detects a requested event as a Request, then the SSF shall send a EDP-Request message, and shall populate the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of “FALSE.”

sap *(Sap)*:

When the originating access is an SS7 trunk and the received IAM contains the *Sap* parameter, the SSF shall map the *Sap* parameter received in the IAM directly to the corresponding *Sap* parameter in the INAP message.

7.4.6.9 Origination Attempt

When the SSF sends this information to the SCF, it should use the same rules as for AnalyzedInformation, with the following exceptions:

serviceAddressInformationServiceKey *(UserID)*:

The serviceAddressInformationServiceKey *(UserID)* parameter shall contain the identity of the originating facility. A DN included in this parameter shall be the calling party number of the facility, not its billing number.

chargeNumber *(ChargeNumber)*:

The *chargeNumber (ChargeNumber)* parameter shall contain the Automatic Number Identification (ANI) of the originating facility. This value shall be a 10-digit NANP number. If the originating access is a non-ISDN line or an ISDN interface, then the SSF shall include the *chargeNumber (ChargeNumber)* parameter for the non-ISDN line or ISDN interface.

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to “offHookImmediate”.

callingPartysCategory *(ChargePartyStationType)*:

callingPartysCategory *(ChargePartyStationType)* in the Origination_Attempt message shall contain the OLI[3] of the originating non-ISDN line or ISDN interface. The callingPartysCategory *(ChargePartyStationType)* shall be coded using the Class of Service information that shall be mapped to the appropriate value.

carrier *(Carrier)*:

The carrier *(Carrier)* parameter consists of two fields that are populated as follows:

> The Carrier Selection field indicates whether the primary carrier was presubscribed or dialed. This field in the Origination_Attempt message shall indicate that the primary carrier was presubscribed.
>
> The Carrier ID field shall contain the presubscribed carrier.

7.4.6.10 O Suspended

When the SSF sends this information to the SCF, it should use the same rules as for OAnswer.

7.4.6.11 O Term Seized

When the SSF sends this information to the SCF, it should use the same rules as for OAnswer.

7.4.6.12 Route Select Failure

When the SSF sends this information to the SCF, it should use the same rules as for AnalyzedInformation, with the following exceptions:

serviceAddressInformationServiceKey *(UserID)*:

The serviceAddressInformationServiceKey *(UserID)* parameter shall contain the identity of the originating facility. The serviceAddressInformationServiceKey *(UserID)* may be a DN (e.g., for non-ISDN interfaces), ISDNI, TrunkGroupID (for public trunks), or PrivateFacilityGID (for private trunks). A DN included in this parameter shall be the calling party number of the facility, not necessarily its billing number. If a call previously encountered call forwarding, the serviceAddressInformationServiceKey *(UserID)* is the identity of the last forwarding station.

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to “aFR”.

7.4.6.13 T Answer

When the SSF sends this information to the SCF, it should use the same rules as for OAnswer.

7.4.6.14 T Busy

When the SSF sends this information to the SCF, it should use the same rules as for TerminationAttempt, with the following exceptions:

serviceAddressInformationServiceKey *(UserID)*:

If the SSF detects the T_Busy trigger, the SSF shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the identity of the terminating facility or DN to which the trigger is assigned.

If the SSF detects a requested event as an EDP-Request, then the SSF shall send an EDP-Request message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to “tBusy”.

busyCause *(BusyCause)*:

The SSF populates the busyCause *(BusyCause)* parameter according to the rules for OCalledPartyBusyArg, except for the call rejected events, since these events do not cause the SSF to encounter the T_Busy DP.

busyType *(BusyType)*:

If the SSF is able to offer the call to the called party using the OfferCall message, it shall populate the *BusyType* parameter with a value of “callCanBeOffered”. Otherwise, the SSF shall populate the *BusyType* parameter with a value of “callCannotBeOffered”.

serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:

If the SSF detects a requested event as a Request, then the SSF shall send a EDP-Request message, and shall populate the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of “FALSE.”

sap *(Sap)*:

When the originating access is an SS7 trunk and the received IAM contains the *Sap* parameter, the SSF shall map the *Sap* parameter received in the IAM directly to the corresponding *Sap* parameter in the INAP message.

7.4.6.15 Termination Attempt

When the SSF sends this information to the SCF, it should use the same rules as for AnalyzedInformation, with the following exceptions:

serviceAddressInformationServiceKey *(UserID)*:

The serviceAddressInformationServiceKey *(UserID)* parameter shall contain the DN to which the Termination Attempt trigger is assigned.

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to "terminationAttempt".

calledPartysCategory *(CalledPartyStationType)*:

The calledPartysCategory *(CalledPartyStationType)* contains the OLI[3] of the called party.

When the called party is served by the SSF, the SSF shall include the calledPartysCategory *(CalledPartyStationType)* parameter in the message. The calledPartysCategory *(CalledPartyStationType)* shall be coded using the called party's Class of Service information, which shall be mapped to the appropriate value.

displayInformationGN *(GenericName)*:

If the originating access is an SS7 trunk, the SSF shall include the TCAP displayInformationGN *(GenericName)* parameter when the ISUP generic name and calling party number parameters are available in the received IAM.

If the originating access is a non-ISDN line or an ISDN BRI interface, the SSF shall include the displayInformationGN *(GenericName)* parameter if the calling party has invoked either Calling Name Delivery Blocking (CNAB) or Calling Identity Delivery and Suppression (CIDS).

If the originating access is an ISDN PRI, the SSF shall include the displayInformationGN *(GenericName)* parameter if the calling party has invoked Presentation of Calling Name (PCN).

7.4.6.16 Timeout

When the SSF sends this information to the SCF, it should use the same rules as for O No Answer.

7.4.6.17 T No Answer

When the SSF sends this information to the SCF, it should use the same rules as for TerminationAttemptArg, with the following exceptions:

serviceAddressInformationServiceKey *(UserID)*:

If the SSF encounters the T_No_Answer trigger, then the SSF shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the identity of the terminating facility or DN to which the trigger is assigned.

If the SSF detects a requested event as an EDP-Request, then the SSF shall send an EDP-Request message, and shall populate the serviceAddressInformationServiceKey *(UserID)* parameter with the same value that was sent with the TDP-Request message that initiated the transaction.

serviceAddressInformationTriggerType *(TriggerCriteriaType)*:

The serviceAddressInformationTriggerType *(TriggerCriteriaType)* parameter shall be set to "tNoAnswer".

sap *(Sap)*:

When the originating access is an SS7 trunk and the received IAM contains the *Sap* parameter, the SSF shall map the *Sap* parameter received in the IAM directly to the corresponding *Sap* parameter in the INAP message.

serviceAddressInformationMiscCallInfo *(NotificationIndicator)*:

If the SSF detects a requested event as a Request, then the SSF shall send a EDP-Request message, and shall populate the serviceAddressInformationMiscCallInfo *(NotificationIndicator)* parameter with a value of “FALSE.”

7.4.7 Services assumed from TCAP

7.4.7.1 Normal procedures

This clause describes procedures that are to be followed for transmitting messages between the SSF, SCF and SRF to support T1 IN functionality under normal operation. TCAP (Transaction Capability Application Part) transactions are used to send the information between the SSF, SCF and the SRF.

The transaction portion defines procedures used to begin, continue and end transactions between functional entities.

The component portion defines procedures used to exchange operations as an ordered sequence of components.

The dialog portion defines optional procedures that are only required if an application context needs to be exchanged between functional entities.

7.4.7.1.1 Definitions

Categories of messages sent between the SSF, SCF and SRF, and information regarding TCAP Transactions are described in the following clauses. Call-related messages are messages related to call processing. Abnormal messages are not directly call-related messages nor non-call-related messages.

7.4.7.1.1.1 Switch call-related messages

A switch call-related message is a call-related message sent from the SSF to the SRF or SCF. For T1 IN, the only types of switch call-related messages that are sent to the SCF or SRF are switch request messages and switch notification messages.

7.4.7.1.1.1.1 Switch request message

A switch request message is a switch call-related message that indicates that an SCF or SRF response message is required. Switch request messages include trigger detection point request messages (TDP-R’s) and event detection point request messages (EDP-R’s).

7.4.7.1.1.1.2 Switch response message

A switch response message is a switch call-related message that indicates that an SCF or SRF response is required and is sent in reply to a particular SCF or SRF call-related message or messages. A switch response message is correlated to the SCF or SRF message to which it replies. Switch response messages include Call_Info_From_Resource and ResourceClear.

7.4.7.1.1.1.3 Switch notification message

A switch notification message is a switch call-related message that indicates that no corresponding SCF or SRF response message should be sent. Switch notification messages include event detection point notification messages (EDP-N's) and Close.

7.4.7.1.1.2 SCF/SRF call-related message

An SCF or SRF call-related message is a call-related message sent from the SCF or SRF to the SSF. An SCF or SRF call-related message requests the SSF to execute functions (tasks).

7.4.7.1.1.2.1 SCF/SRF response message

An SCF or SRF response message is an SCF/SRF call-related message that must be sent after a switch request message or after a switch conversation message is received. SCF/SRF response messages are correlated to switch call-related messages. SCF/SRF response messages will close the transaction.

A TCAP unidirectional message could also be sent after the transaction has been closed.

7.4.7.1.1.2.2 SCF/SRF event message

An SCF or SRF event message is an SCF/SRF call-related message sent to the SSF asynchronously as a unidirectional message, rather than sent in reply to a switch request message. An SCF/SRF event message cannot be sent prior to sending an SCF/SRF response message, when an SCF/SRF response message is required. Cancel_Resource_Event is an SCF/SRF event message.

7.4.7.1.1.3 SCF/SRF non-call-related message

An SCF or SRF non-call-related message is a non-call-related message sent from the SCF or SRF to the SSF. An SCF/SRF non-call-related message cannot modify the state of a call. Monitor_For_Change is an example of an SCF/SRF non-call-related message.

7.4.7.1.1.4 Switch non-call-related message

A switch non-call-related message is a non-call-related message sent from the SSF to the SCF or SRF. A switch non-call-related message is either sent in reply to an SCF/SRF non-call-related message or is initiated by the switch. For example, Termination_Notification is the switch non-call-related message sent in reply to the SCF/SRF Send_Notification request, in which the equivalent of a transaction level correlation is done via the EchoData parameter received from the SCF/SRF in the Send_Notification message. NCA_Data is an example of a switch non-call-related message initiated by the switch.

7.4.7.1.1.5 Multiple-component package

A multiple-component package is a TCAP package that contains more than one TCAP component. In T1 IN, this can only occur when an SCF/SRF response message and at least one SCF/SRF non-call-related message are sent in a single TCAP package.

7.4.7.1.1.6 Transactions

7.4.7.1.1.6.1 Transaction already exists

The following terms are used when a transaction already exists and a TCAP package is sent:

- Maintained - Query was done without permission, or conversation was established without permission (transaction should not be closed).
- Not Maintained - Query was done with permission, or conversation was established with permission, or response was sent (transaction may be closed).

7.4.7.1.1.6.2 Transaction does not already exist

The following terms are used when a transaction does not already exist and a TCAP package is sent:

- Query - Transaction should be established.
- Unidirectional - Transaction should not be established.

7.4.7.1.1.6.3 Call-related transaction

A call related transaction is a transaction that is established by the sending of a call-related message.

7.4.7.1.1.6.4 Non-call-related transaction

A non-call-related transaction is a transaction that is established by the sending of a non-call-related message.

7.4.7.2 Abnormal procedures

This clause describes procedures that shall be followed for transmitting information between the SSF and then SCF or SRF to support T1 IN functionality under abnormal conditions. The ANSI TCAP protocol is used to send the information between the SSF and the SCF/SRF. These procedures are divided into protocol errors, application errors, failures and caller abandon. Refer to T1.114 for additional information regarding the error procedures.

Annex A defines constraints on the use of the TCAP protocol by INAP. If a constraint is violated, the ASE detecting the constraint violation reports it according to the above-mentioned procedure using the reject problem code specified by the constraint.

7.4.7.2.1 Definitions

This clause describes the terms associated with abnormal procedures.

1. Fatal errors - Fatal errors are conditions that cause an operation to be unsuccessful and may cause a transaction to be closed. The only fatal errors that do not cause a transaction to be closed are fatal application errors detected in the ACG or Furnish_AMA_Information message.
2. Non-fatal errors - Non-fatal errors are error conditions that do not interfere with the completion of an operation.
3. Protocol errors - Protocol errors are caused by incorrect TCAP packages.
4. Application errors - Application errors are errors that are associated with a received TCAP package that violate the requirements associated with sending the TCAP package and are of a nonprotocol nature.
5. Failures - A failure occurs when a hardware or software resource within the SCF, SSF or SRF, is unable to provide functionality that is necessary for the completion of an operation requested by a message.

7.4.7.2.2 Protocol errors

The SSF detects protocol errors in TCAP packages received from the SCF or SRF. The SCF/SRF detects protocol errors in TCAP packages received from the SSF. Protocol errors can occur in the transaction portion, the component portion or the dialogue portion of a TCAP package.

7.4.7.2.3 Application errors

The SSF shall detect application errors in messages received from the SCF or SRF. The SCF or SRF shall detect application errors in messages received from the SSF.

For additional information on application errors, please refer to 7.4.2.1.1.

7.4.7.2.4 Failures

A failure occurs when hardware or software resources within the SSF, SCF or SRF are unable to provide functionality necessary for completion of a requested operation. The types of failures that may be reported by the SSF, SCF or SRF are listed in the ASN.1 description of the *FailureCause* parameter.

Annex A

(normative)

A Remote Operations Information Objects and Syntax

Annex A maps T1 INAP on to ANSI TCAP. The mapping is consistent with the formats and codes of T1.114.

```
IN–Remote–Operations–Information–Objects {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0)
informationObjects(5) version1(0) }
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 {
      local                             INTEGER,
      global                            OBJECT IDENTIFIER,
      national          [PRIVATE 16]    IMPLICIT INTEGER -32768..32767,
      private           [PRIVATE 17]    IMPLICIT INTEGER
      }

ErrorCode ::= CHOICE {
      local                             INTEGER,
      global                            OBJECT IDENTIFIER,
      national          [PRIVATE 19]    INTEGER  -128..127,
      private           [PRIVATE 20]    INTEGER
      }

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

IN–TCAP–ROS–PDUs {iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) generic–ROS–PDUs(6)
version1(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
--exports everything
IMPORTS OPERATION, ERROR FROM
{ iso(1) memberbody(2) us(840) t1-667-2002(10069) modules(0) informationObjects(5) version1(0)};
ROS { InvokeId: InvokeIdSet, OPERATION: Invocable, OPERATION: Returnable } ::= CHOICE {
      invokeLast            [PRIVATE 9]     Invoke {{InvokeIdSet}, {Invocable}},
      (CONSTRAINED BY { -- invocable.&invokeLast must be TRUE -- }
      ! RejectProblem : general–incorrectComponentPortion),
      returnResultLast      [PRIVATE 10]    ReturnResult {{Returnable}},
      returnError           [PRIVATE 11]    ReturnError {{Errors{{Returnable}}}},
      reject                [PRIVATE 12]    Reject
      invokeNotLast         [PRIVATE 13]    Invoke {{InvokeIdSet}, {Invocable}},
      (CONSTRAINED BY { -- invocable.&invokeLast must be FALSE -- }
      ! RejectProblem : general–incorrectComponentPortion),
      returnResultNotLast   [PRIVATE 14]    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)
                                            – 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,
   opcode                           OPERATION.&operationCode
                                    ((Operations)
                                    ! RejectProblem : invoke–unrecognisedOperation ),
   argument                         OPERATION.&ArgumentType
                                    ((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 T1.114.3 encoding rules  -- }
! RejectProblem : general–incorrectComponentCoding )

ReturnResult { OPERATION: Operations } ::= SEQUENCE {
   correlationId       [PRIVATE 15]  IMPLICIT OCTET STRING SIZE(1)
   (CONSTRAINED BY { -- must be that for an outstanding operation -- }
   ! RejectProblem : returnResult–unrecognisedCorrelationId)
   (CONSTRAINED BY { -- which returns a result -- }
   ! RejectProblem : returnResult–unexpectedReturnResult),
   result                            OPERATION.&ResultType
   ({Operations}{@opcode})
   ! RejectProblem : returnResult–incorrectParameter)
   }

(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 T1.114.3 encoding rules  -- }
! RejectProblem : general–incorrectComponentCoding )

ReturnError { ERROR: Errors } ::= SEQUENCE {
   correlationId       [PRIVATE 15]  IMPLICIT OCTET STRING SIZE(1),
   (CONSTRAINED BY ( -- must be that for an outstanding operation-- )
   ! RejectProblem : returnError–unrecognisedCorrelationId)
   (CONSTRAINED BY { -- which returns an error-- }
   ! RejectProblem : returnError–unexpectedReturnError),
   errCode                           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 T1.114.3 encoding rules  -- }
```

```
! RejectProblem : general–incorrectComponentCoding )

Reject ::= SEQUENCE {
    correlationID       [PRIVATE 15]  IMPLICIT OCTET STRING SIZE(0..1),
    problem             [PRIVATE 21]  IMPLICIT RejectProblem,

    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 T1.114.3 encoding rules  -- }
! RejectProblem : general–incorrectComponentCoding )

RejectProblem ::= 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)
    }

END -- end of generic ROS PDU& definitions
```

Annex B

(informative)

B SCF-SDF interface

B.1 Introduction

The SCF-SDF interface is defined for use on both intranetwork and internetwork interfaces. The SCF-SDF interface is defined in the ITU-T Recommendation Q.1218 (1995).

B.2 Alignment between the ITU-T Recommendation X.500 concepts and the IN

This interface is based on the ITU-T Recommendation X.500 that are used to specify the SCF-SDF interface and the contents of the SDF.

Most of the concepts of the ITU-T Recommendation X.500 Recommendations are directly used in the IN environment. When looking at the structure of the SCF, the Service Data Management is the part of the SCF responsible for the interactions with the SDF. It can be mapped onto the concept of Directory User Agent (DUA). The SDF is the entity responsible for answering database requests. This functional entity can be mapped onto the Directory System Agent (DSA). When an association is set up between an SCF and a SDF, an instance of a DSA is created for the length of the association. The Directory is a collection of DSAs/SDFs. This set can be used for a specific service or for a variety of services. The notion of Directory is equivalent to the concept of database systems in IN.

B.3 Use of a limited subset of the ITU-T Recommendation X.500

The primary purpose of the ITU-T Recommendation X.500 is to provide a directory service. However, the functionalities defined cover more than the functionalities needed for IN. Profiling is used as a means to present the status of the different parameters. For convenience and clarity, this profile is defined using Abstract Syntax Notation (ASN.1) subtyping facilities. However, these definitions do not form a protocol specification; instead, the profile indicates specific parameters an implementation should not send. This does not change the behavior of the receiving entity that shall still be capable of decoding values that conform to the original definition of the Directory Abstract Service. Nevertheless, elements that are excluded by subtyping should be ignored.

B.4 The IN Directory Access Protocol (DAP) subset

The IN Directory Access Protocol (DAP) subset is defined in the ITU-T Recommendation Q.1218 (1995). Information types and common procedures, including operations for Bind, Search, AddEntry, RemoveEntry, and ModifyEntry, are defined in this clause. Errors are defined for Attribute, Service, Security, and Update in this clause. The DAP protocol subset and ITU-T Recommendation X.501 profile are also described. The full ASN.1 profiling of the Directory Abstract Service is provided. Finally, the ASN.1 type and value definitions contained in the Directory Specification are provided in the ASN.1 module, “DirectoryAccessProtocol”.

B.5 Using the SCF-SDF interface in the North American environment

The SCF-SDF interface is defined for use only with ITU-T TCAP. In order to provide an SCF-SDF interface definition for the North American environment, it is necessary to map the SCF-SDF interface definition from ITU-T TCAP to ANSI TCAP. It is expected that this mapping will be provided in subsequent versions of this specification.

The SCF-SDF interface definition may be used for international internetworking interfaces. The necessary interworking between ITU-T TCAP and ANSI TCAP to provide such an interface is not defined in this specification. It is expected that this interworking description will be provided in subsequent versions of this specification.

Annex C

(informative)

C Bibliography

ITU-T Recommendation Q.1200 (03/93); *Q-Series intelligent network Recommendation structure*[2]

ITU-T Recommendation I.312 / Q.1201 (10/92); *Principles of intelligent network architecture*[2]

ITU-T Recommendation I.328 / Q.1202 (10/92); *Intelligent network - Service plane architecture*[2]

ITU-T Recommendation I.329 / Q.1203 (10/92); *Intelligent network - Global functional plane architecture*[2]

ITU-T Recommendation Q.1204 (3/93); *Intelligent network - Distributed functional plane architecture*[2]

ITU-T Recommendation Q.1205 (3/93); *Intelligent network - Physical plane architecture*[2]

ITU-T Recommendation Q.1208 (3/93); *General aspects of the intelligent network application protocol*[2]

ITU-T Recommendation Q.1211 (3/93); *Introduction to intelligent network capability set 1*[2]

ITU-T Recommendation Q.1213 (10/95); *Global functional plane for intelligent network CS-1*[2]

ITU-T Recommendation Q.1214 (10/95); *Distributed functional plane for intelligent network CS-1*[2]

ITU-T Recommendation Q.1215 (10/95); *Physical plane for intelligent network CS-1*[2]

ITU-T Recommendation Q.1218 (10/95); *Interface Recommendation for intelligent network CS-1*[2]

ITU-T Recommendation Q.1219 (4/94); *Intelligent network users guide for capability set 1*[2]

ITU-T Recommendation Q.1290 (10/95); *Glossary of terms used in the definition of intelligent networks*[2]

ITU-T Recommendation X.500 (11/93) | ISO/IEC 9594-1:1993, *Information technology - Open Systems Interconnection - The directory: Overview of concepts, models and services*[2]

ITU-T Recommendation X.501 (11/93) | ISO/IEC 9594-2:1993, *Information technology - Open Systems Interconnection -The directory: Models*[2]

ITU-T Recommendation X.509 (11/93) | ISO/IEC 9594-8:1993, *Information technology - Open Systems Interconnection -The directory: Authentication framework*[2]

ITU-T Recommendation X.511 (11/93) | ISO/IEC 9594-3:1993, *Information technology - Open Systems Interconnection -The directory: Abstract service definition*[2]

ITU-T Recommendation X.518 (11/93) | ISO/IEC 9594-4:1993, *Information technology - Open Systems Interconnection -The directory: Procedures for distributed operation*[2]

ITU-T Recommendation X.519 (11/93) | ISO/IEC 9594-5:1993, *Information technology - Open Systems Interconnection -The directory: Protocol specifications*[2]

GR-1129-CORE Issue 5, November, 2000; *AINGR: Switch – Intelligent Peripheral Interface (IPI)*[4]

GR-1298-CORE Issue 6, November, 2000; *AINGR: Switching Systems*[4]

GR-1299-CORE Issue 6, November, 2000; *Advanced Intelligent Network (AIN) Switch – Service Control Point (SCP)/Adjunct Interface*[4]

[4] Telcordia documents are available from Industry Direct Sales, Telcordia, 8 Corporate Place, PYA 3A-184, Piscataway, NJ, 08854-4156, or: < http://telecom-info.telcordia.com >.

Annex D

(informative)

D Note on Informative References

The information contained in this annex provides additional detail for the Transaction and Component Level procedures.

Annex C contains a list of Informative References for supplemental information. Refer to these references for additional detail regarding the procedures.

The current INAP Transaction and Component Level Procedures may not be described in sufficient detail to assure interoperability of INAP implementations. By including these informative references, the additional Transaction and Component Level procedural detail will facilitate interoperability.

For example, where this T1 IN standard may not provide sufficient detail to implement the OMidCall procedure, then Section 3, “Normal Procedures” and Section 4, “Abnormal Procedures” of Telcordia Technologies GR-1299-CORE describe the Transaction and Component Level Procedures in further detail and may be used as a reference to facilitate interoperability. These Normal and Abnormal Procedures Sections would apply to all procedures specified in this document.

As a second example, where this T1 IN standard may not provide sufficient detail to implement the AMA parameter procedures, then Section 9, “Automatic Message Accounting” of Telcordia Technologies GR-1298-CORE describes the AMA Parameter Procedures in further detail and may be used as a reference to facilitate interoperability. This AMA Parameter Procedures Section would apply to all procedures specified in this document.

As a third example, where this T1 IN standard may not provide sufficient detail to implement the DPNumber parameter mapping, then Section 4, “SSP Processing of Triggers and Requested Events” of Telcordia Technologies GR-1298-CORE describes the Detection Points and a possible mapping.

Annex E

(informative)

E Source Material for T1 IN

The T1 IN standard is based on ITU-T IN CS-1 and IN CS-2 recommendations and on *Telcordia Technologies AIN Generic Requirements* (GR-1298-CORE and GR-1299-CORE). The T1 IN standard incorporates provisions to meet the unique needs of the North American telecommunications environment and to support service standardization initiatives of Committee T1 requiring standardized IN capabilities and signaling.

Table E-1 indicates the source material for the T1 INAP. In most instances, the operations are AIN-based (GR-1299-CORE) (with several name changes as noted to align with ITU-T terminology); otherwise, the operations are ITU-T INAP-based (ITU-T Recommendation Q.1218). For example, the T1 IN "analyzedInformation" operation is based on the GR-1299-CORE "infoAnalyzed" operation, but the name of the operation was changed; the T1 IN "analyzeRoute" operation is based on the GR-1299-CORE "analyzeRoute" operation; and the T1 IN "activateServiceFiltering" operation is based on the ITU-T "activateServiceFiltering" operation.

Table E-1 - T1 IN Application Protocol

T1 IN Operation	***AINAP GR-1299 based***	***Equivalent GR-1299 name***	***ITU INAP based***
acq	√		
activateServiceFiltering			√
activityTest			√
analyzedInformation	√	***infoAnalyzed***	
analyzeRoute	√		
applyCharging			√
applyChargingReport			√
assistRequestInstructions			√
authorizeTermination	√		
callInfoFromResource	√		
callInformationReport			√
CallInformationRequest			√
cancelResourceEvent	√		
close	√		
collectedInformation	√	***infoCollected***	
collectInformation	√		

T1 IN Operation	***AINAP GR-1299 based***	***Equivalent GR-1299 name***	***ITU INAP based***
continue	√		
eventNotificationCharging			√
facilitySelectedAndAvailable	√	***termResourceAvailable***	
forwardCall	√		
furnishChargingInformation			√
instructionsToSRF			√
monitorForChange	√		
monitorSuccess	√		
nCAData	√		
nCARequest	√		
oAnswer	√		
oCalledPartyBusy	√		
oDisconnect	√		
oDTMFEntered	√		
offerCall	√		
oMICall			√
oNoAnswer	√		
originationAttempt	√		
oSuspended	√		
oTermSeized	√		
releaseCall	√	***disconnect***	
reportError	√		
requestNotificatinChargingEvent			√
requestReportBCMEvent	√		
resetTimer			√
resourceClear	√		
routeSelectFailure	√	***networkBusy***	
sendChargingInformation			√
sendNotification	√		
sendToResource	√		
serviceFilteringResponse			√

T1 IN Operation	***AINAP GR-1299 based***	***Equivalent GR-1299 name***	***ITU INAP based***
statusReported	√		
tAnswer	√		

tBusy	√		
terminationAttempt	√		
teminationNotification	√		
timeout	√		
tMidCall			√
tNoAnswer	√		
update	√		
updateRequest	√		