INTERNATIONAL TELECOMMUNICATION UNION

ITU-T Q.1229

(03/99)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

SERIES Q: SWITCHING AND SIGNALLING

Intelligent Network

Intelligent Network user's guide for Capability Set 2

ITU-T Recommendation Q.1229

(Previously CCITT Recommendation)

ITU-T Q-SERIES RECOMMENDATIONS

SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4 AND No. 5	Q.120–Q.249
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	**Q.1200–Q.1699**
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700–Q.1799
BROADBAND ISDN	Q.2000–Q.2999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION Q.1229

INTELLIGENT NETWORK USER'S GUIDE FOR CAPABILITY SET 2

Summary

This Recommendation is intended to provide a detailed guide for the capabilities provided by Intelligent Network Capability Set 2 (INCS-2). This guide includes examples of service scenarios, as well as details to ensure an understanding and to assist in the implementation of IN CS-2. This guide is targeted towards a wide audience that includes Users that only require a general "talking" knowledge on IN and how it will be utilized, as well as Users that need a detailed "working" knowledge of IN in order to complete their work function within an IN structured environment. This Recommendation is accompanied by one annex and two appendices.

Source

ITU-T Recommendation Q.1229 was prepared by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on 15 March 1999.

FOREWORD

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

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

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation the term *recognized operating agency (ROA)* includes any individual, company, corporation or governmental organization that operates a public correspondence service. The terms *Administration, ROA* and *public correspondence* are defined in the *Constitution of the ITU (Geneva, 1992)*.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Introduction

Intelligent Network Capability Set 2 (IN CS-2) is the second standardized capability set of the IN.

The phased approach of capability sets has been described in Recommendation Q.1201 and has been shown in Recommendation Q.1211 (Refer to Figure 1/Q.1211).

General description and scope of IN CS-2 are described in clause 3/Q.1221.

This Recommendation is intended to help users who will implement or use IN CS-2 functionality. For this purpose, this Recommendation provides an overview of the specification for each IN CS-2 key function, useful and detailed information not described in the other IN CS-2 Recommendations, and also several service scenario examples.

Capabilities out of the scope of IN CS-2 are not addressed in this Recommendation; however, this Recommendation includes some material closely related to Intelligent Network Capability Set 3 (IN CS-3) because studies of some IN CS-3 services/service features started in the IN CS-2 time-frame.

This Recommendation is aligned with the framework of Recommendation Q.1219 (Intelligent Network User's Guide for Capability Set 1) and Q.1219 Supplement (Intelligent Network User's Guide: Supplement for IN CS-1) and focuses on the service, network and management aspects newly introduced in IN CS-2.

Recommendation Q.1229

INTELLIGENT NETWORK USER'S GUIDE FOR CAPABILITY SET 2

(Geneva, 1999)

1 Scope

1.1 Target audience

The Target Audience includes a broad spectrum of IN Users of this guideline. At one end of the spectrum is a set of Users that only require a general "talking" knowledge of IN and how it will be utilized. At the other end of the spectrum is a set of Users that need a detailed "working" knowledge of IN in order to complete their work function within an IN structured network. Specifically, this guideline is written for use by Service Providers and Equipment Vendors (as described in Recommendation Q.1201/I.312) and by Manufacturers and Network Operators (as described in Q.1201/I.312).

These needs within the Target Audience can be satisfied with this User's Guide as discussed in 1.2, Intended use.

This Recommendation does not contain any information described in the Recommendation Q.1219 IN CS-1 User's Guide and only focuses on the IN CS-2 specific aspects. Therefore readers are assumed to have full knowledge of IN CS-1 to understand the detailed parts of this Recommendation.

1.2 Intended use

The intent of this IN CS-2 User's Guide is to provide general and detailed guideline for implementation of IN capabilities provided in IN CS-2 which is the second standardized stage of the IN as an architectural concept for the creation and provision of services, including Telecommunications services, Service Management services and Service Creation services.

This User's Guide is intended to be used:

a) as a delta document based on the IN CS-1 User's Guide in the sense described in the previous subclause;

b) as a reference document to understand the relationship of IN CS-2 to the IN Conceptual Model (Recommendation Q.1201/I.312), to the previous phase (IN CS-1) and next phase (IN CS-3), and to the target architecture;

c) to direct the User who requires more detailed information on the specific service, network and management aspects that are not fully described in the IN CS-2 series of Recommendations, i.e. the Q.122X-series Recommendations;

d) to provide example service scenarios to help the users understand the usage of IN CS-2 Recommendations.

1.3 Organization

Clause 1 introduces the scope of this Recommendation.

Clause 2 contains the list of references.

Clause 3 defines terminology used in this Recommendation.

Clause 4 introduces the capabilities provided for IN CS-2.

General descriptions for the IN CS-2 service aspects and architectural aspects are provided in clauses 5 and 6 respectively. In particular, clause 6.3 briefly describes the role of each plane of the IN Conceptual Model (INCM) and describes IN CS-2 specific aspects in each plane.

Clause 7 describes the IN CS-2 infrastructure which follows the INCM principles. Subclause 7.1 provides the overview of the IN CS-2 specification items necessary to realize the IN CS-2 key functions and 7.2 provides detailed and useful information for IN CS-2 users not described in the other IN CS-2 Recommendations.

Annex A provides Service Scenario Examples as illustrations of the previously described IN CS-2 capabilities.

1.4 Framework outline of the Q.1200-series Recommendations

The following table, taken from clause 1/Q.1200, provides the IN Recommendation structure:

Q.12n0	**Q.12nX**
00 – General	1 – Principles introduction
n0 – CS-n (1...8)	2 – Service plane
90 – Vocabulary	3 – Global functional plane
	4 – Distributed functional plane
	5 – Physical plane
	6 – For future use
	7 – For future use
	8 – Interface Recommendation
	9 – Intelligent Network user's guide

The IN CS-2 Recommendations (Q.122X series) form a detailed and stable basis for achieving implementation of IN CS-2 telecommunication services. IN CS-2 Recommendations are intended to give the same degree of technical information as for the IN CS-1 Recommendations (1995).

In order to prepare the next phase of IN (i.e. IN CS-3), the IN CS-2 Recommendations contain both complete technical specifications (with complete support for physical architecture and detailed protocol description) and incomplete specifications that are intended to be the base for IN CS-3 study. The latter include some services description and a part of Distributed Functional Plane (DFP) specifications.

The following is a summary of the IN CS-2 series of Recommendations.

Q.1220: Q.1220-series Intelligent Network Capability Set 2 Recommendation Structure

This Recommendation gives the structure for all IN CS-2 Recommendations.

Q.1221: Introduction to Intelligent Network Capability Set 2

This Recommendation gives an introduction to IN CS-2 by providing an overview and definition of IN CS-2 and by describing its main characteristics and overall capabilities. It defines the service aspects, network aspects and functional relationships that form the basis of the IN CS-2 capabilities.

The following Q.122X-series Recommendations are based on the general framework provided in Recommendation Q.120X, consistent with the scope of IN CS-2 defined in Recommendation Q.1221.

Q.1222: Service Plane for Intelligent Network Capability Set 2

This Recommendation provides the architecture of the IN CS-2 Service Plane of the INCM such that specific functionalities and their interactions can be identified and described in other Recommendations.

Q.1223: Global Functional Plane for Intelligent Network Capability Set 2

This Recommendation provides the functional characteristics of the Global Functional Plane (GFP) architecture of the INCM for IN CS-2 including:

– the IN GFP model for IN CS-2;

– IN CS-2 Service Independent Building Blocks (SIBs) (stage 1) including the specialized SIBs, Basic Call Process (BCP) and Basic Call Unrelated Process (BCUP);

– the mapping of the Service Plane to the Global Functional Plane.

Q.1224: Distributed Functional Plane for Intelligent Network Capability Set 2

This Recommendation describes the DFP of the INCM for IN CS-2 including:

– the IN DFP architecture for IN CS-2 with static and dynamic models of the Functional Entities (FEs) related to IN service execution;

– the IN DFP call model and service processing model for IN CS-2;

– the IN DFP SIB stage 2 descriptions to identify Information Flows and Functional Entity Actions;

– the IN DFP detailed Information Flow descriptions, including Information Elements and functional descriptions, as the basis for specifying IN protocols.

Q.1225: Physical Plane for Intelligent Network Capability Set 2

This Recommendation describes the Physical Plane of the IN architecture for CS-2. The Physical Plane of the IN CS-2 identifies the different Physical Entities (PEs) and the interfaces between these entities. This Recommendation provides typical example scenarios of FE to PE mapping.

Q.1228: Intelligent Network Interface for Capability Set 2

This Recommendation defines the Intelligent Network Application Protocol (INAP) for the support of capabilities required by the IN CS-2 target services over the IN CS-2 interfaces as defined in Recommendation Q.1221. Recommendation Q.1228 defines the Application Protocol Data Units (APDUs) which are used between the Physical Entities and the procedures to be followed by each functional entity to provide services.

Q.1229: Intelligent Network User's Guide for Capability Set 2

This Recommendation is intended to provide both general and detailed guideline for the implementation of the IN capabilities provided in IN CS-2. It includes example service scenarios, as well as details to insure an understanding and implementation of IN CS-2.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- CCITT Recommendation I.312/Q.1201 (1992), *Principles of intelligent network architecture*.
- CCITT Recommendation I.328/Q.1202 (1992), *Intelligent network – Service plane architecture*.
- ITU-T Recommendation Q.71 (1993), *ISDN circuit mode switched bearer services*.
- ITU-T Recommendation Q.704 (1996), *Signalling network functions and messages*.
- ITU-T Recommendation Q.708 (1993), *Numbering of international signalling point codes*.
- ITU-T Recommendation Q.711 (1996), *Functional description of the signalling connection control part*.
- ITU-T Recommendation Q.713 (1996), *Signalling connection control part formats and codes*.
- ITU-T Recommendation Q.771 (1993), *Functional description of transaction capabilities*.
- ITU-T Recommendation Q.772 (1993), *Transaction capabilities information element definitions*.
- ITU-T Recommendation Q.773 (1993), *Transaction capabilities formats and encoding*.
- ITU-T Recommendation Q.1200 (1993), *Q-series intelligent network Recommendation structure*.
- ITU-T Recommendation Q.1204 (1993), *Intelligent network distributed functional plane architecture*.
- ITU-T Recommendation Q.1211 (1993), *Introduction to intelligent network Capability Set 1*.
- ITU-T Recommendation Q.1215 (1993), *Physical plane for intelligent network CS-1*.
- ITU-T Recommendation Q.1219 (1994), *Intelligent network user's guide for Capability Set 1*.
- ITU-T Recommendation Q.1219 Supplement 1 (1997), *Intelligent network user's guide: Supplement for IN CS-1*.
- ITU-T Recommendation Q.1220 (1997), *Q.1220-series intelligent network Capability Set 2 Recommendation structure*.
- ITU-T Recommendation Q.1221 (1997), *Introduction to intelligent network Capability Set 2*.
- ITU-T Recommendation Q.1222 (1997), *Service plane for intelligent network Capability Set 2*.
- ITU-T Recommendation Q.1223 (1997), *Global functional plane for intelligent network Capability Set 2*.
- ITU-T Recommendation Q.1224 (1997), *Distributed functional plane for intelligent network Capability Set 2*.
- ITU-T Recommendation Q.1225 (1997), *Physical plane for intelligent network Capability Set 2*.
- ITU-T Recommendation Q.1228 (1997), *Interface Recommendation for intelligent network Capability Set 2*.
- ITU-T Recommendation X.500 (1993) | ISO/IEC 9594-1:1995, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services.*

– ITU-T Recommendation X.509 (1993) | ISO/IEC 9594-8:1995, *Information technology – Open Systems Interconnection – The Directory: Authentication framework.*

– ITU-T Recommendation X.525 (1993) | ISO/IEC 9594-9:1995, *Information technology – Open Systems Interconnection – The Directory: Replication.*

3 Vocabulary

This clause provides the lists of IN CS-2 key words, terminology and abbreviations, used in this Recommendation.

3.1 Terms and definitions

This subclause provides a list of IN CS-2 specific terminology used in this Recommendation. Terminology shown here relate with new aspects and/or new concepts of IN CS-2. Appropriate references are also indicated.

Basic Call Unrelated Process (BCUP) specialized SIB: 6.2/Q.1223

Basic Call Unrelated State Model (BCUSM): 8.2.1/Q.1224

Basic Service Management Process specialized SIB: Appendix I/Q.1223

Call Party Handling (CPH): see 7.1.4

Call Segment (CS): 4.3.1 and 4.3.3/Q.1224

Call Segment Association (CSA): 4.3.1 and 4.3.3/Q.1224

Call Unrelated Service Function (CUSF): 3.3.8/Q.1224 and clause 8/Q.1224

Capability View: clause 4/Q.1223

Chaining: 12.5.3.2.5 and 14.4.2.2/Q.1228

Connection Point (CP): 4.3.1/Q.1224

Connection View State (CVS) approach: 4.3.3/Q.1224

Controlling leg: 4.3.1/Q.1224

Controlling SCF: 12.5.3.1/Q.1228

IN CS-2 key functions: 7.2/Q.1221

Distributed service logic: 11.5/Q.1224

Distribution of service logic: 3.4.2/Q.1224

Domain: clause 4/Q.1223

Enhanced SRF: see 7.1.5

Entry method: see 7.2.3.9

Generic security mechanism: see 7.1.11

High Level SIB (HLSIB): clause 4/Q.1223

Hybrid approach: 4.3.4/Q.1224

IN-SM core capabilities: 4.3.2/Q.1224

Internetworking between IN structured network: see 7.1.9

Internetworking between non-IN structured network: see 7.1.10

ISDN CPE: 5.1 and 5.3.11/Q.1225

Leg status: 4.3.1/Q.1224

Mid-Call interruption: see 7.1.3

Out-Channel Call Related User Interaction (OCCRUI): see 7.1.7

Out-channel Call Unrelated User Interaction (OCUUI): see 7.1.6

Parallel service processing: 4.1.2.3/Q.1223

Passive leg: 4.3.1/Q.1224

Personal Mobility: 7.2.14/Q.1221

Point In Activation (PIA): 8.2.2/Q.1224

Point Of Synchronization (POS): clause 4/Q.1223

Referral: 16.1.4, 16.1.12 and 16.1.14/Q.1228

SCF-IAF interface: see 7.1.10

SCF-SCF interface: see 7.1.9

SDF-SDF interface: see 7.1.9

Service Call Unrelated Service Point (CUSP): clause 3, 5.1 and 5.3.11/Q.1225

Service Control User Agent Function (SCUAF): 3.3.9/Q.1224

Service Creation service/service feature: Appendix I/Q.1221

Service Management service/service feature: Appendix I/Q.1221

Service Process: clause 4/Q.1223

Service View: clause 4/Q.1223

Shadowing: 14.4.2.1/Q.1228

SIB operation: clause 4/Q.1223

Supporting SCF: 12.5.3.1/Q.1228

Telecommunications Management Network (TMN) concept: Annex B/Q.1224

Terminal mobility: see 7.1.13

User Interaction-Scripts: 3.4.5 and 3.4.6/Q.1224

3.2 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

APDU Application Protocol Data Unit

BCP Basic Call Process

BCSM Basic Call State Model

BCUP Basic Call Unrelated Process

BCUSM Basic Call Unrelated State Model

CCAF Call Control Agent Function

CCF Call Control Function

CP Connection Point

CPH Call Party Handling

CRACF	Call related Radio Access Control Function
CS	Call Segment
CS	Capability Set
CSA	Call Segment Association
CSM	Call Segment Model
CURACF	Call Unrelated Radio Access Control Function
CUSF	Call Unrelated Service Function
CVS	Connection View State
DAP	Directory Access Protocol
DFP	Distributed Functional Plane
DIB	Directory Information Base
DIT	Directory Information Tree
DN	Directory Number
DN	Distinguished Name
DP	Detection Point
DSA	Directory System Agent
DSP	Directory System Protocol
DSS1	Digital Subscriber Signalling No. 1 Protocol
DTMF	Dual Tone Multi Frequency
DUA	Directory User Agent
ECMA	European Computer Manufacturers Association
EDP	Event Detection Point
EDP-N	Event Detection Point-Notification
EDP-R	Event Detection Point-Request
FE	Functional Entity
FEA	Functional Entity Action
FIM	Feature Interactions Manager
FSM	Finite State Machine
GFP	Global Functional Plane
GSL	Global Service Logic
GVNS	Global Virtual Network Services
HLSIB	High Level Service Independent Building Block
IAF	Intelligent Access Function
IE	Information Element
IF	Information Flow
IMT-2000	International Mobile Telecommunications-2000
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
ISUP	Integrated Services Digital Network-User Part
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
MACF	Multiple Association Control Function
OCUUI	Out-channel Call Unrelated User Interaction
PIA	Point In Activation
PIC	Point In Call
PM	Personal Mobility
POI	Point Of Initiation
POR	Point Of Return
POS	Point Of Synchronization
PRI	Primary Rate Interface
PSTN	Public Switched Telephone Network
RCF	Radio Control Function
RCP	Resource Control Part
RM	Resource Manager
ROSE	Remote Operations Service Element
SACF	Single Association Control Function
SCCP	Signalling Connection Control Part
SCEF	Service Creation Environment Function
SCF	Service Control Function
SCME	Service Control Function Management Entity
SCSM	SCF Call State Model
SCUAF	Service Control User Agent Function
SDF	Service Data Function
SDL	Specification and Description Language
SF	Service Feature
SIB	Service Independent Building Block
SL	Service Logic
SLP	Service Logic Processing Program
SLPI	Service Logic Processing Program Instance
SMAF	Service Management Access Function

SMF Service Management Function

SRF Specialized Resource Function

SRSM SRF call State Model

SS7 Signalling System No. 7

SSF Service Switching Function

SSN Subsystem Number

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

UI-Scripts User Interaction-Scripts

UPT Universal Personal Telecommunication

4 Capabilities provided by Capability Set 2

IN CS-2 is a superset of IN CS-1. The objectives for IN CS-2 are the following:

– Although, by nature, the IN is a service-independent architecture, it is relevant to describe the general IN CS-2 service capabilities. IN CS-2 provides network capabilities defined to support the set of IN CS-2 benchmark services and service features (refer to clause 5/Q.1221). These capabilities can also be used for the support of other services/service features that may not be standardized by ITU-T.

– IN CS-2 provides some functionality to evolve towards succeeding IN Capability Sets (IN CS-3 and beyond), in particular towards terminal mobility aspects – International Mobile Telecommunications-2000 (IMT-2000), service management and service creation service aspects.

Recommendation Q.1201 provides guidance on general aspects such as service implementation independence, multi-vendor capability, multi-network capability, rapid service delivery and service deployment. IN CS-2 capabilities cover all these general aspects. As in IN CS-1, the IN CS-2 architecture may be supported by, but is not limited to, PSTN, ISDN and mobile networks.

As in IN CS-1, IN CS-2 focuses on normal call processing scenarios. Issues related to error handling and exception handling are outside the scope of IN CS-2 studies for the upper planes of the INCM. However, general information about error cases of DP processing at an SSF/CCF and subsequent actions which should be taken to handle these errors can be found in 4.2.8/Q.1224. Subclause 4.2/Q.1228 specifies several error types for the IN CS-2 INAP, and clause 16/Q.1228 defines general procedures for error handling for these error types.

5 Service aspects for IN CS-2

IN CS-2 proposes benchmark services/service features to identify the network capabilities that should be supported by IN CS-2 structured networks. The set of IN CS-2 benchmark services/service features is a superset of the set of IN CS-1 benchmark services/service features.

The IN CS-2 benchmark services/service features are categorized into the following three groups: Telecommunication services/service features, Service Management services/service features, and

Service Creation services/service features. However, IN CS-2 Recommendations do not provide a complete set of specifications for all of these services/service features. In terms of terminal mobility related services/service features which belong to the first group and all service/service features belonging to the other two groups, only a part of Distributed Functional Plane (DFP) specifications are defined. The complete set of specifications regarding these services/service features will be provided in future IN capability sets.

The set of IN CS-2 benchmark services/service features are shown in Appendix I/Q.1221.

5.1 Telecommunication service aspects

IN CS-2 enables a network operator or a service provider to introduce various types of telecommunication services/service features which are not supported by an IN CS-1 structured network. Examples of these new types of telecommunication services/service features are:

- internetworking services/service features (e.g. Internetwork Freephone service, Internetwork Premium Rate service, etc.);
- personal mobility services/service features (e.g. User Authentication, User Registration, UPT service optional service features, etc.); and
- Call Party Handling (CPH) services/service features (e.g. Call Transfer, Call Waiting, etc.).

IN CS-2 Recommendations define all necessary specifications for IN CS-2 benchmark telecommunication services/service features including the above services/service features.

However, in terms of terminal mobility related services/service features (e.g. Terminal Authentication, Handover, etc.), IN CS-2 Recommendations only specify a part of the required functional architecture of IN networks. The DFP architecture for the terminal mobility is defined by introducing new Functional Entities and relationships between them. This aspect is identified as a normative part of the IN CS-2 specifications and is described in Annex A/Q.1224. Information flows and information elements are also defined for the terminal mobility; however, these are not identified as a normative part of the IN CS-2 specifications but are identified as an informative part. Information flows and information elements for the terminal mobility are described in Appendix II/Q.1224. Additional specifications necessary for terminal mobility related services/service features will be provided in IN CS-3 or in later capability sets.

As it was the case for IN CS-1, IN CS-2 capabilities are intended to support services/service features that fall into the category of "single ended", "single point of control" services.

An IN CS-2 structured network provides users the following telecommunication service aspects extended from those for IN CS-1.

- **Flexible routing**: There is no significant enhancement on this basic capability. An IN CS-2 structured network enables a service provider to maintain control of routing decisions as in an IN CS-1 structured network. These routing decisions may be based on time-of-day, day-of-week, authorizations codes, etc. The routing decision criteria will be managed by the service provider. IN CS-2 extends the call model specified in IN CS-1 to provide additional routing flexibility.
- **Flexible charging**: There is no significant enhancement on this aspect. Charging decisions in an IN CS-2 structured network may be under the control of the service provider. As in IN CS-1, charging mechanisms can be based on locations, destinations, authorization codes, etc. On the basis of IN CS-1 capabilities, charging scenarios are provided in Appendix II/Q.1214 with the related information flows and information elements.
- **Flexible user interaction**: As in IN CS-1, the capability to support user interaction for a specific service is provided to the service provider. IN CS-2 enhances this aspect in various ways. IN CS-1 restricts an IN structured network so that it can interact with a user only through an in-channel connection. An IN CS-2 structured network can interact with a user

not only through an in-channel connection but also through an out-channel signalling connection during a call. In addition to this, an IN CS-2 structured network can also interact with a user through an out-channel signalling connection when the user is not involved in a call.

- **Multi-party control**: An IN CS-2 structured network enables a network operator or a service provider to create CPH services/service features which involve three or more parties within a single service instance, while IN CS-1 restricts up to only two parties being involved within a single service instance. The basic technique used to realize this multi-party control capability is the "Connection View State (CVS) approach" which is based on the idea of controlling multiple basic calls in an SSF/CCF at the same time from a service logic instance in an SCF. Moreover, a network operator or a service provider can create more complicated services/service features than those based on CVS approach by using a "hybrid approach" which is defined based on the CVS approach. The hybrid approach uses bridging functions provided by an SRF. These CPH services/service features can be invoked during the active phase of a call.
- **Service interaction**: In IN CS-1, general rules have been specified for basic call modelling. In IN CS-2, new service interaction processing capabilities are provided. New rules are specified for the call unrelated aspects. The content of the service interaction indicator is defined whereas it is network specific in IN CS-1. However the complete mechanism for the service interaction detection and resolution are not specified in IN CS-2.
- **Service interworking over network boundaries**: IN CS-2 encompasses new service aspects for the purpose of interworking over several networks that require technical and commercial agreements between network operators. In addition to IN CS-1 SCF-SDF relationship, IN CS-2 supports SCF-SCF and SDF-SDF relationships for the telecommunication service processing where the relationships across between two IN structured networks. Distributed service logic and distributed data processing are realized by using these relationships. SCF-IAF relationship is also identified for interworking between an IN structured network and a non-IN structured network although no protocol specifications are provided especially for this relationship. The protocol specifications defined for the SCF-SCF relationship may be also used for this relationship. Generic security mechanism is defined for secured access over network boundaries.

These service aspects are realized by making use of network key functions described in 7.1.

5.2 Service management service aspects

Service management services/service features are listed as a part of IN CS-2 benchmark services/service features whereas these types of service requirements were completely out of the scope of IN CS-1. In order to meet these services/service features requirements, IN network service management functional architecture was investigated and a part of them was specified in IN CS-2 Recommendations. The rest of specification items including protocol specifications on these aspects are not included in IN CS-2 Recommendations but will be studied and specified in later capability sets.

These service aspects focus on the network operator's management activities such as service deployment, service provisioning, and service management. After the deployment, service customization services, service control services and service monitoring services will be used in the provisioning and utilization phases.

While IN CS-1 focused exclusively on the service invocation phase, IN CS-2 defines and links all phases within the scope of service management issues.

5.3 Service creation service aspects

These service aspects focus on the network operator's service creation activities such as service specification, service development and service verification, and were out of the scope of IN CS-1.

Although service creation services/service features are listed as a part of IN CS-2 benchmark services/service features, only a part of functional architecture for service creation services/service features is standardized in IN CS-2.

6 IN CS-2 architecture

6.1 Functions

The network functions addressed in IN CS-2 are grouped into the following categories. The two former are enhanced from the IN CS-1 network functions and the others are newly introduced to IN CS-2.

Call control related functions

IN CS-2 call control related functions are similar to those for IN CS-1 but the capabilities of the Service Switching Function (SSF), Call Control Function (CCF) and the Specialized Resource Function (SRF) are enhanced to realize the IN CS-2 key functions such as Mid-call interruption, Call Party Handling, Enhanced SRF, Out-Channel Call Related User Interaction, Service/Feature Interaction, and Terminal Mobility, all of which are described in 7.2/Q.1221.

Refer to subclauses 3.1 to 3.3/Q.1224 for the description of distributed functional architecture and each functional entity (FE). Subclause 7.1 summarizes the IN CS-2 specifications reflecting those enhancements for each key function.

Terminal Mobility requires introduction of new FEs in addition to the above FEs to handle mobile calls. These are Call-related Radio Access Control Function (CRACF), Call-Unrelated Radio Access Control Function (CURACF) and Radio Control Function (RCF). Refer to Annex A/Q.1224 for these new FEs.

NOTE – Terminal Mobility related DFP architecture specifications are defined in Annex A independently from the other IN CS-2 DFP architecture related specifications.

Service control related functions

For a single IN CS-2 structured network, the service control related functions within the context of a call supported by the Service Control Function (SCF) and the Service Data Function (SDF) are extended to realize the IN CS-2 key functions such as Call Party Handling, Enhanced SRF, Out-Channel Call Related User Interaction, Service/Feature Interaction and Terminal Mobility.

In addition to the above enhancement, the internetworking service control related functions are introduced when multiple IN CS-2 structured networks are involved in a call. Multiple IN CS-2 structured networks can interwork together through not only the SCF-SDF interface (specified in IN CS-1) but also the SCF-SCF and the SDF-SDF interfaces. Refer to 7.1.9 for the internetworking between IN structured networks.

Intelligent Access Function (IAF) is introduced for service control of a non-IN structured network from the SCF in an IN CS-2 structured network. Refer to 7.1.10 for additional information on this function.

Call unrelated user interaction functions

IN CS-2 introduces a new network function for "user interaction". This new function is introduced to perform user interaction not only within the context of a call but also outside the context of a call, while an IN CS-1 structured network can only perform user interaction within the context of a call.

New FEs, CUSF (Call Unrelated Service Function) and SCUAF (Service Control User Agent Function) are defined to realize this function.

The CUSF is responsible for handling a call unrelated relationship with the SCUAF through the out-channel interface and also for handling a relationship with the SCF. The CUSF provides a call unrelated event processing mechanism to detect a request from a user for interacting with the SCF and also performs procedures as required from the SCF for call unrelated user interaction.

The SCUAF represents the functions for the interface between a user and the CUSF. The SCUAF enables a user to interact with the CUSF for call unrelated user interaction.

Refer to 7.1.6 for additional information on this function.

Management related functions

Service Management Function and Service Creation Function are included in the scope of IN CS-2. Three new FEs, Service Management Function (SMF), Service Management Access Function (SMAF) and Service Creation Environment Function (SCEF) are introduced for these management functions.

The SMF provides various kinds of the service management functions for each phase of service management, including service deployment, service provisioning and service utilization. Telecommunication Management Network (TMN) concepts are utilized as a basis for identifying the service management activities and modelling the IN CS-2 network elements for these management activities.

The SMAF provides an interface function which enables a user to access the SMF. The "user" here means a service subscriber or a service administrator.

The SCEF provides supporting functions of "service specification" phase, "service development" phase and "service verification" phase of service creation activities.

Refer to 7.1.14, 7.1.15 and 7.1.16 for management-related functions.

6.2 Functional relationships and interfaces

The relationships between functional entities for the DFP of an IN CS-2 structured network identified for service control and management activities are as follows:

IN CS-2 relationships

SCF-SSF, SCF-SCF, SCF-IAF, SCF-SRF, SDF-SDF, SCF-SDF, SCF-CUSF, SMF-SCF, SMF-SDF, SMF-SSF/CCF, SMF-SRF, SMF-SMAF, SMF-SCEF, SMF-SMF, SMF-CUSF.

Refer to clause 7/Q.1221 and to 3.4/Q.1224 for these relationships.

6.2.1 Internetworking in IN CS-2

The following relationships are identified for internetworking in IN CS-2:

IN CS-2 relationships for internetworking

SCF-SCF, SCF-IAF, SDF-SDF, SCF-SDF, SMF-SMF.

In IN CS-2, specific internetworking capabilities are assumed to be localized within the FEs supporting the internetworking relationships, i.e. within the SCF and the SDF. The internal architecture of a network is not visible. However, functions needed for processing internetworking are to be visible from the other network (IN or non-IN).

The SCF-SDF relationship has been defined in IN CS-1 and it has already provided a part of internetworking capabilities. The new relationships SCF-SCF and SDF-SDF provide different capabilities for internetworking from those provided by the SCF-SDF relationship. These new

relationships enable the requesting network to be free from understanding the details of service logic, data schema or data location in the requested network.

By using the SCF-SCF interface, two service logics can communicate with each other. This interface allows the distribution of service logic. A network can handle a call without having a full knowledge of the data schema and the service logic as long as it can find another network that can help.

The SDF-SDF internetworking interface has two purposes: the first is to provide a mechanism to copy data between a network and a maintenance of the copy data, and the second is to provide transparent data access. The following requirements pertain to this relationship: security, performance merit for no additional load to the SCF, data location management, copy of data, information updated, and data transparency.

For the propose of internetworking, the SMF-SMF relationship has been identified as within the scope of IN CS-2, but no information flows nor information elements have been defined in IN CS-2.

For the SCF-IAF relationship, the same information flows and information elements as used for the SCF-SCF relationship may be used.

6.2.2 Other functional relationships and interfaces

In order to assist in the efficient development and use of the capability set, 3.4/Q.1224 provides a description of the use of the IN CS-2 relationships required to support the target set of services and service features.

Service control related relationships SCF-SSF, SCF-SRF and SCF-SDF were specified in IN CS-1 and are evolved in IN CS-2 to realize the IN CS-2 key functions described in 7.2/Q.1221.

The SCF-CUSF relationship is defined in IN CS-2 and used for the out-channel call unrelated user interaction. The CUSF may also have relationships with SSF and CCF but standardization of these relationships are not in the scope of IN CS-2. The relationship between CUSF and SCUAF is not the subject of IN standardization activity and existing appropriate protocols such as DSS1 should be adopted for this interface by the network provider.

In IN CS-2, the relationships SCF-CURACF and SCF-CRACF are identified to support the terminal mobility function. Information flows and information elements regarding these new relationships and modification of the SCF-SSF relationship for the terminal mobility function are provided in Appendix II/Q.1224 as a part of the informative parts of the IN CS-2 specifications.

Various relationships for the management activities, from SMF to other FEs, are identified in IN CS-2 but no information flows nor information elements are defined. Subclauses 3.4.13 to 3.4.20/Q.1224 outline the management activities supported by the SMF for other FEs.

6.3 IN CS-2 INCM aspects

The INCM is described in Recommendation Q.1201. The subclauses below describe the IN CS-2 INCM aspects.

For IN CS-2, some Services or Service Features are studied and described only in the two or three upper planes and will be completely specified in IN CS-3 or later capability sets.

6.3.1 IN CS-2 Service Plane

IN CS-1 did not address the service plane because IN CS-1 was developed based on the existing network evolution into the IN concepts. IN CS-2 provides the first view of the "top-down" approach where services and service features are initially defined and then the network capabilities necessary to realize these services and service features are developed in the lower planes of the IN Conceptual Model.

IN CS-2 also addresses feature interactions which were not included in IN CS-1; in particular, consideration on the methods to identify service and service feature interaction are explained in 2.3/Q.1222.

A structured approach, as described in Recommendation Q.1202, is applied to analyse services and to decompose services into service features.

Service plane modelling is described in 2.4/Q.1222.

Clause 7/Q.1223 entitled "Mapping of the Service Plane to the Global Functional Plane", describes how a service feature in Service Plane is mapped to the GFP.

6.3.2 IN CS-2 Global Functional Plane

IN CS-2 specifies two different views of the GFP: a "Capability view" and a "Service view". Each view describes different aspects of the GFP. The "Capability view" identifies the set of basic network capabilities based on Service Independent Building Blocks (SIBs) and the discrete SIB operations concept.

Other aspects of the GFP are described by a "Service view". "Service view" shows how global service logic is composed of SIBs, SIB operations and "High level SIBs" which are service independent re-usable components. "Service view" also describes how global service logic interrelate to each other in parallel service processing. Refer to 7.1.17 for more details about these views.

SIBs can be considered as tools to identify information flows, information elements and functional entity actions in the DFP. These are obtained by elaborating SIBs in the form of "SIB stage 2 description". Subclause 11.2/Q.1224 provides "SIB stage 2 description" for every IN CS-2 SIB.

In terms of a service control scheme in the GFP, the basic framework is the same as that of IN CS-1. A global service logic interacts with a specialized SIB for the basic call (i.e. BCP SIB) through the interaction points specified for both the global service logic and the BCP SIB. In addition, a new specialized SIB named "Basic Call Unrelated Process (BCUP)" is defined for the out-channel call unrelated interaction function. The BCUP SIB interacts with global service logic through the interaction points in the case of the out-channel call unrelated interaction processing. Subclause 11.3/Q.1224 provides the stage 2 descriptions for these specialized SIBs.

IN CS-2 is also intended to provide management capabilities on the GFP so as to support management services; however, specifications have not been fully provided. Modelling of the management activities on the GFP consists in enhancing existing SIBs and in creating a specialized SIB, "Basic Service Management Process" SIB. Some guidance for this modelling approach is provided in Appendix I/Q.1223.

6.3.3 IN CS-2 Distributed Functional Plane

The IN DFP architecture encompasses static and dynamic models of the Functional Entities (FEs) related to IN service execution. These models are used to define how an IN service logic instance interacts with basic call process. The DFP for IN CS-2 is a subset of the general DFP described in Recommendation Q.1204.

The IN DFP detailed information flow descriptions, including information elements and functional descriptions are the basis for specifying IN Application Protocol (INAP). Most of the information flows and information elements defined in IN CS-2 DFP are mapped to the INAP operations and their parameters. However, no protocol specifications for the terminal mobility, service management and service creation related functions are provided in IN CS-2. Protocol aspects for these functions are out of the scope of IN CS-2 and will be defined in IN CS-3 or later capability sets.

Subclause 3.1.5/Q.1228 indicates the mapping of the IN CS-2 information flows to IN CS-2 INAP operations.

Telecommunication Management Network (TMN) concepts and protocol specifications are assumed to be used for the management related functions in future capability sets. Refer to Annexes B, C and D of Recommendation Q.1224 for the service management aspects.

6.3.4 IN CS-2 Physical Plane

This plane identifies the Physical Entities (PEs) and protocols and indicated the mapping of FEs to PEs. New PEs, Call Unrelated Service Point (CUSP) and ISDN CPE are defined according to the introduction of new FEs in IN CS-2 DFP. The CUSP contains CUSF and CCF, and ISDN CPE may contain SCUAF, IAF and CCAF. An example protocol architecture for the CUSF and SCUAF is shown in 3.1.1/Q.1228.

The mapping of the IN CS-2 terminal mobility, service management and service creation related FEs are out of the scope of IN CS-2.

7 Infrastructure in IN CS-2

This clause provides overview of IN CS-2 specifications and helpful information not covered in other IN CS-2 Recommendations.

Main points of the IN CS-2 specifications for realizing each IN CS-2 network key function are summarized under each subclause of 7.1.

Useful information for IN CS-2 users not covered or clearly described in other CS-2 Recommendations (e.g. general guidelines, example cases or detailed protocol aspects) are described in 7.2.

7.1 Overview of IN CS-2 specifications

The following IN CS-2 specific network key functions (or capabilities) are identified to achieve IN CS-2 objectives, service aspects described in clause 5.

1) Single Point/Multipoints of Control;
2) Single-Ended/Multi-Ended Calls;
3) Mid-Call interruption;
4) Call Party Handling;
5) Enhanced SRF;
6) Out-channel Call Unrelated User Interaction;
7) Out-Channel Call Related User Interaction;
8) Service/Feature Interaction (Service Processing);
9) Internetworking between IN structured networks;
10) Internetworking with non-IN structured networks;
11) Security;
12) Personal Mobility;
13) Terminal Mobility;
14) IN-TMN;
15) Service Management;
16) Service Creation.

The first thirteen functions are mainly concerned with IN CS-2 Telecommunication services and service features. The last three are concerned with IN CS-2 Service Management and Service Creation service and service features. Definitions of these key functions are described in 7.2/Q.1221.

These functions are regarded as functional requirements to be supported by the IN CS-2 structured network. IN CS-2 specifications are defined to meet these functional requirements.

Relationships between each IN CS-2 network key function and IN CS-2 specification items are briefly described in 7.1.1 to 7.1.16. A summary of IN CS-2 GFP specifications not directly related with IN CS-2 network key functions is provided in 7.1.17.

7.1.1 Single Point of Control/Multipoints of Control

Single Point of Control describes a control relationship where the same aspects of a call are influenced by one and only one Service Control Function at any point in time. Multiple Points of Control is the ability for multiple service instances to interact with a single call segment. The SSF/CCF may have to manage interactions between IN service logic instances realized in different service logic instances that are simultaneously active on a single call.

IN CS-2 is still restricted to the "Single Point of Control" rule.

Some clarification to the scope of the "Single Point of Control" has been provided in IN CS-2 timeframe (see 4.2.8/Q.1224): the DP processing rules only guarantee single point of control within a Single Control Relationship. Within a single SSF/CCF, many control relationships may exist but it must be made clear that single point of control only relates to each single relationship.

7.1.2 Single-Ended/Multi-Ended Calls

In IN CS-1, a service logic instance in an SCF can control only one "half-call" part of a call in an SSF/CCF. This "single-ended service feature" principle is extended in IN CS-2 so that a service logic instance can also control associated "half-calls" or a multi-party "half-call". In any case, only a single controlling party is involved in the IN control. This extension is made for supporting Call Party Handling function described in 7.1.4. (Refer to 4.2/Q.1219 for illustrative examples of Single-Ended Service Feature and also Single Point of Control concepts. The last part of 4.3.1/Q.1224 describes the IN CS-2 extension.)

7.1.3 Mid-Call Interruption

Mid-Call interruption is the functionality to allow the existing Mid-Call TDPs to function beyond the case in IN CS-1, enabling a user to invoke an IN service or service feature during the active phase of a call. IN CS-2 specifications supporting this function are summarized in the following:

GFP specification items

- BCP SIB has enhanced POIs (Points of Initiation) and PORs (Points of Return) according to the IN CS-2 requirement to allow more interaction between the basic call and the service logic. "Call Interrupted" POI is enhanced for Mid-Call Interruption. (Refer to 6.1.2.1/Q.1223).

DFP specification items

- The following five trigger types are identified for mid-call interruption:
  - O_Switched_Hook_Flash_Immediate;
  - O_Switched_Hook_Flash_Specific_Code;
  - T_Switched_Hook_Flash_Immediate;
  - T_Switched_Hook_Flash_Specific_Code;
  - BRI_Feature_Activation_Indicator.

  The first four trigger types are for switch-hook flash and the last one is for feature activation indication.

- When a TDP is met and its trigger type is one of trigger types listed above, SSF/CCF is requested to process this TDP appropriately to send an SCF necessary information with O_MidCall or T_MidCall information flow (refer to 4.2.7/Q.1224).

  The requested process to SSF/CCF includes the following:

  – Switch-based non-IN call control function is required to hold a passive party during mid-call interruption, and to offer dial tone to the controlling party and collect digits from the controlling party. A passive party denotes here a party not requesting the mid-call interruption and a controlling party denotes a party requesting the mid-call interruption (refer to 7.1.4 and to 4.3.2.1/Q.1224).

  – SSF/CCF must be able to interpret the digits provided by the controlling party to determine how the successive call processing is treated (e.g. IN is involved or not, which service feature is required, etc.) (refer to 4.3.2.1/Q.1224).

- Component, Component Correlation ID and Component Type information elements are added to the list of information elements for O_MidCall and T_MidCall information flows in order to provide an SCF with information received from the controlling party (refer to 12.4.3.45, 12.4.3.71, 12.4.4.33, 12.4.4.34 and 12.4.4.35/Q.1224).

Protocol specification items

- Component, componentCorrelationID and componentType optional parameters are added to the "MidCallArg" (refer to 5.1/Q.1228).
- In IN CS-2, SCF Call State Model (SCSM) in terms of SSF and SRF interface (SCSM-SSF/SRF) contains various kinds of FSMs (refer to 12.5.1/Q.1228). "FSM for CS", one of these FSMs, is considered to allow receiving MidCall EDPs in any state in it (refer to 12.5.1.3/Q.1228).

7.1.4 Call Party Handling

Call Party Handling is the ability to manage various parties participation in a call. IN CS-2 call modelling/processing aspect is enhanced so as to allow parties bearer channels to be added, deleted, joined and/or separated from the other parties involved in the call. IN CS-2 specifications supporting this function are summarized as in the following:

GFP specification items

- JOIN and SPLIT SIBs are newly defined in IN CS-2 for CPH. JOIN, SPLIT and BCP SIBs are used within the context of CPH in the GFP (refer to 5.8, 5.15 and 6.1/Q.1223).

DFP specification items

- The object-oriented technique is used to describe IN Switching State Model (IN-SSM), where the key concept is Connection View State (CVS). CVS represents the state of basic calls and their related connections maintained by the SSF/CCF by using connection view objects such as Call Segment Association (CSA), Call Segment (CS), Legs, Connection Point (CP) and BCSM (refer to 4.3/Q.1224).

  – **CSA**: provides an SCF with an abstract view of a single two-party or a multi-party call segment, or of a pair of associated call segments. The CSA 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.

  – **CS**: represents a "half-call" part of a two-party call (either originating or terminating part) or a "half-call" part of a multi-party call.

  – **CP**: represents a joint function between two legs, a conference function between three or more legs, replication function, merging function, 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 IN CS-2, it interconnects legs supported by equivalent bearer services, and supports interworking between circuit mode/speech and circuit mode/3.1 kHz audio bearer services.

– **Leg**: is typed as either a "controlling leg" or a "passive leg". A controlling leg represents the local access interface at a local exchange or the remote access interface at transit exchange (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 signalling (e.g. a mid-call event) or on behalf of an end user. A passive leg represents a communication path which receives indications from the other side of the interface of the call, not the controlling side. "Leg status" such as "null", "pending", "joined", "shared" and "surrogate" are specified according to the status and context of the related call(s).

- The "CVS approach" is introduced to realize service features involving more than three parties, where the basic idea is controlling multiple basic calls and related connections in the SSF/CCF from the SCF (refer to 4.3.1/Q.1224 for IN-SSM and to 4.3.3/Q.1224 for the CVS approach).
- The "Hybrid approach", designed based on the CVS approach, is also introduced to realize more complicated services than those realized by the CVS approach. The hybrid approach uses bridging function at an SRF (refer to 4.3.4/Q.1224).
- IN Switching Manager (IN-SM) is required to enhance its capability for CPH . Four IN-SM core capabilities are identified for IN CS-2 CPH processing:
  – capability for Mid-Call Interruption as shown in the previous subclause;
  – capability to connect to a resource/transfer each call party;
  – capability to present the current view of the half call and connection state to the SCF; and
  – capability to combine selected transferred paths into a single call,

  (Refer to 4.3.2/Q.1224 for more detail about IN-SM core capabilities for CPH).
- As multiple BCSMs are involved in CPH processing, event detection/reporting rules concerning which BCSM is responsible for handling the event and how the event is notified, are necessary. Precedence rules among related BCSMs for the detection/reporting of events signalled on the controlling leg are specified (refer to 4.3.3.6/Q.1224).
- Fourteen CVSs are identified for IN CS-2 CPH processing as shown in the following (refer to 4.3.3.7/Q.1224). Note that this set of CVSs does not contain every possible state represented by a combination of CV objects but is considered as a set of typical examples of such possible states:
  – **Null**: represents a condition where call processing is not active. There is no controlling leg or passive leg connected to the connection point.
  – **Originating Setup**: represents an originating two-party call in the set-up phase.
  – **Stable 2-party**: represents a stable or clearing two-party call, and is either an originating or a terminating call from the perspective of the controlling user.
  – **Terminating Setup**: represents a terminating two-party call in the set-up phase.
  – **M-party Setup**: represents two associated call segments: one for an originating two-party call in the set-up phase and the other for either an originating or a terminating two-party call in the stable or clearing phase. The controlling user only has put one party on hold, and has originated a new call which has not yet reached the stable phase. Note

that how the passive user can put one party on hold for some services in order to receive the request from the remote party side at the transit exchange is for further study.

- **Call on Hold**: represents two associated call segments: one for a two-party call in the stable or clearing phase and the other for either an originating or a terminating two-party call in the stable or clearing phase. The controlling user only has put one party on hold, and is participating in another call which is in the stable or clearing phase. Note that how the passive user can put one party on hold for some services in order to receive the request from the remote party side at the transit exchange is for further study.
- **Call Waiting**: represents two associated call segments: one for a two-party call in the terminating set-up phase in call waiting and the other for either an originating or a terminating two-party call in the stable or clearing phase. The controlling user is participating in a call that is in the stable or clearing phase, and another call is terminating to the controlling user.
- **Stable M-Party**: represents a stable or clearing multi-party call in one call segment.
- **Transfer**: represents a transferred call. CS in this CVS contains a controlling leg in "Surrogate" status and two passive legs in "Joined" status. The call between the two passive legs is in the stable or clearing phase. Note that the "surrogate" legStatus for the controlling leg indicates the charging relationship between the two passive legs after the call has been transferred.
- **Forward**: represents a forwarded call. Call processing for the first passive leg is in a stable or clearing phase, or a terminating call set-up phase, whereas call processing for the second passive leg is in an originating call set-up phase.
- **Originating Setup M-Party**: represents two associated call segments, both for originating two-party call in the set-up phase. The controlling user is in the set-up phase (e.g. connected to an SRF for bridging and the SRF has originated a new call which has not yet reached the stable state).
- **Active M-Party**: represents two associated call segments: one for an originating two-party call in the set-up phase and the other for an originating two-party call in the stable phase.
- **1-Party Setup**: represents a 1-party call being originated on behalf of the network (i.e. the controlling leg has a legStatus = surrogate).
- **Stable 1-Party**: represents a 1-party call originated on behalf of the network (i.e. the controlling leg has a legStatus = surrogate) that is in a stable or clearing phase.

• Fourteen information flows (IFs) from an SCF to an SSF shown in the following are identified for IN CS-2 CPH processing. Seven of these IFs are newly introduced in IN CS-2, and are summarized in Table 7-1. These information flows cause CVS transitions as described in 4.3.3.7/Q.1224.

- AnalyseInformation;
- CollectInformation;
- CreateCallSegmentAssociation;
- CreateCallSegmentAssociationResult;
- Connect;
- DisconnectLeg;
- InitiateCallAttempt;
- MergeCallSegments;
- MoveLeg;
- MoveCallSegments;

– Reconnect;
– ReleaseCall;
– SplitLeg;
– SelectRoute.

The hybrid approach does not require all of CVSs and IFs shown above and uses a subset of them (refer to 4.3.4.3/Q.1224).

Table 7-1/Q.1229 – New IFs/IEs for CPH (from SCF to SSF)

IF	IEs	Note
Create Call Segment Association (12.4.3.22/Q.1224) <17.37/Q.1228>	Call ID (M)	This IF is used to create a new CSA. The new CSA will not contain any Call Segments after creation. The SSF is responsible for specifying unique CSA identifier for the created CSA.
Create Call Segment Association Result (12.4.3.23/Q.1224)	New Call Segment Association (M)	This IF is used to report the new CSA ID to the SCF. At the operation level, The Return Result of CreateCallSegmentAssociation operation corresponds to this IF.
Disconnect Leg (12.4.3.27/Q.1224) <17.41/Q.1228>	Call ID (M), Leg ID (M), Release Cause (O)	This IF is used to release a specific leg associated with the call and retain any other legs not specified in the Disconnect_Leg IF.
Merge Call Segments (12.4.3.38/Q.1224) <17.62/Q.1228>	Call ID (M), Source Call Segment (M), Target Call Segment (M)	This IF is issued by the SCF to merge two associated CSs with a single controlling leg into one CS with that controlling leg. The net effect of the Merge Call Segment message is to create a communication among the controlling leg and both passive legs, with each party being able to communicate with both other parties.
Move Call Segments (12.4.3.39/Q.1224) <17.64/Q.1228>	Call ID (M), Legs (M), New Call Segment (M), Source Call Segment Association (M), Target Call Segment Association (M)	This IF is used to move a Call Segment from the source Call Segment Association to the target Call Segment Association. This IF ends the association between the moved Call Segment and any Call Segments remaining in the source Call Segment Association.
Move Leg (12.4.3.40/Q.1224) <17.65/Q.1228>	Call ID (M), Leg ID (M), Target Call Segment (M)	This IF is issued by the SCF to move the leg from one CS to another CS with which it is associated. The net effect of the Move Leg message is to interrupt the current communication of the controlling leg, without clearing the passive leg on that communication, and to establish communication for the controlling leg with the other passive leg.

Table 7-1/Q.1229 – New IFs/IEs for CPH (from SCF to SSF) *(concluded)*

IF	IEs	Note
Reconnect (12.4.3.50/Q.1224) <17.82/Q.1228>	Call ID (M), Alerting Pattern (O), Display Information (O), Notification Duration (O)	This IF is used to re-establish communication between the controlling leg and the (held) passive leg(s) of a call with two or more parties, when the controlling leg has disconnected. In particular, this IF requests that BCSM processing set the reconnect timer to the value specified by the Notification Duration IE, and provide the requested Alerting Pattern and/or Display Information to the controlling leg.
Split Leg (12.4.3.66/Q.1224) <17.117/Q.1228>	Call ID (M), Leg ID (M), New Call Segment (M)	This IF is used to separate one party from its Call Segment and, in case of a multi-party CS, place it in a new associated CS. It interrupts the speech connection between the leg to be split and the legs remaining in the original Call Segment. The IF is the inverse of the Merge Call Segments IF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

- For the IFs from SSF to SCF, no new IFs for CPH are specified in IN CS-2. Created Call Segment Association ID Information Element (IE) is defined in order that the SCF can identify a CSA instance in the SSF under the SCF control. This IE is used with DP specific IFs, Initial DP IF and Event Report BCSM (refer to 12.4.4.41/Q.1224).

Protocol specification items

- In IN CS-2, an SSF-FSM instance is classified into one of the following FSMs: "IN-SSM FSM", "Assisting SSF FSM" and "Handed-off SSF FSM". IN-SSM FSM consists of two different FSMs: FSM for CSA and FSM for CS. FSM for CSA creates one or more FSMs for CS synchronized with the CS creation during the CPH processing (refer to 11.3/Q.1228).
- General rules and procedure principles for SSF FSMs in terms of CPH is described in the beginning of 11.5/Q.1228.
- Consideration on SSF/CCF processing for the case it receives "reconnect" operation is shown in 7.2.2.2.2.
- In IN CS-2, SCF Call State Model (SCSM) in terms of SSF and SRF interface contains various kinds of FSMs (refer to 12.5.1/Q.1228). "FSM for CSA" and "FSM for CS" reflect the CPH processing impacts in their state transition (refer to 12.5.1.2 and 12.5.1.3/Q.1228).
- Twelve Call Segment (CS) states represented by a specific combination of Connection View (CV) objects and transitions between these states are specified for IN CS-2 in order to provide strict procedure descriptions of the SSF when CPH related activities take place (refer to A.2 and A.4/Q.1228).

7.1.5 Enhanced SRF

The SRF is enhanced to be able to execute a kind of SRF service logic named "User Interaction-Scripts (UI-scripts)" in order to reduce the number of messages for a series of user interaction procedures (e.g. a procedure for User Authentication) (refer to 3.4.5 and 3.4.6/Q.1224). In addition to that, new types of specialized resources which the SRF controls are added to the existent IN CS-1 specialized resources (these are listed in the following). IN CS-2 specifications supporting this function are summarized as in the following:

GFP specification items

- USER INTERACTION SIB is enhanced to handle UI-scripts. SIB operations of this SIB, "User Interaction RUN", "User Interaction Information" and "User Interaction Close" are defined for this purpose (5.18/Q.1223).

DFP specification items

- SRF internal functional architecture is largely enhanced so as to handle the UI-scripts (refer to clause 5/Q.1224).

  An SRF component RCP (Resource Control Part) is defined for this new capability. RCP has the following functions:

  – **Resource management**: This function is performed by a sub-component "SRF Resource Manager (RM)". The RM allocates an appropriate resource, controls the resource and maintains its status. This function is already identified in IN CS-1.

  – **User Interaction Script execution**: This function is performed by sub-components "User Interaction Scripts (UI-Scripts)", "Transaction Module", "Resource Logic Library" and "Resource Logic Instances". An SCF only requests the execution of a UI-Script to the SRF. Inside the SRF, these sub-components perform a series of user interaction procedure defined by the specified UI-Script on behalf of the SCF and the SRF returns its result to the SCF.

- New types of specialized resources are supported by the IN CS-2 SRF. These are "Automatic Speech Recognition resource", "Text-to-Speech resource" and "Message Sender/Receiver resource" (refer to 3.3.6 and 5.3/Q.1224).
- The SRF-SCF relationship is enhanced so as to support the enhanced SRF capability. As already described above, the SCF does not have to send an operation to the SRF for every user interaction.

  The SRF-SMF relationship is used for the management of the SRF resources.
- Information Flows and Information Elements newly defined for the enhanced SRF capability in IN CS-2 are listed in Tables 7-2 and 7-3:

Table 7-2/Q.1229 – New IFs/IEs for Enhanced SRF (from SCF to SRF)

IF	IEs	Note
Prompt and Receive Message (12.5.2.9/Q.1224) <17.80/Q.1228>	Disconnection From IP Forbidden (M), SRF Connect ID (M), Call Segment (O), Information To Record (O), Information To Send (O), Mailbox ID (O), Media (O), Message Receiving Completion Condition (O), Subscriber ID (O)	This IF is used to receive voice message from call party and record it in voice message sender/receiver. Some announcements are provided if necessary.
Script Close (12.5.2.10/Q.1224) <17.104/Q.1228>	User Interaction Script ID (M), Call Segment (O), User Interaction Script Specific Information (O)	This IF is issued by the SCF to deallocate the resources used to perform the instance of the "User Interaction" script: the context is released.
Script Information (12.5.2.12/Q.1224) <17.106/Q.1228>	User Interaction Script ID (M), User Interaction Script Specific Information (M), Call Segment (O)	This IF is issued by the SCF to send to the SRF additional information during the User Interaction script execution.
Script Run (12.5.2.13/Q.1224) <17.107/Q.1228>	User Interaction Script ID (M), User Interaction Script Specific Information (M), Call Segment (O), Disconnect From IP Forbidden (O)	This IF is issued by the SCF to allocate the necessary resources to perform the instance of the "User Interaction" script and then to activate this "User Interaction" script instance. A context is partially defined for it if necessary.)
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference in Recommendation Q.1228 for the detailed procedure of the operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description, the name of the corresponding operation is the same as that of the IF.		

Table 7-3/Q.1229 – New IFs/IEs for Enhanced SRF (from SRF to SCF)

IF	IEs	Note
Message Received (12.5.2.6/Q.1224)	Received Message ID (M), Received Status (M), SRF Connect ID (M), Received Message Length (O)	This IF is used for confirmation by SCF that the message is received by SRF completely. At the operation level, The Return Result of PromptAnd ReceiveMessage operation corresponds to this IF.
Script Event (12.5.2.11/Q.1224) <17.105/Q.1228>	User Interaction Script ID (M), User Interaction Script Result Information (M), Call Segment (O)	This IF is issued by the SRF to return information to the SCF on the results of the execution of the instance of User Interaction script. This result might be the partial result during the user interaction execution script or the final result of the user interaction script.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Protocol specification items

- State transitions of SRF call State Model (SRSM) according to UI script processing are specified in addition to the existent IN CS-1 transition (refer to 13.4/Q.1228).
- In IN CS-2, SCF Call State Model (SCSM) consists of various kinds of sub FSMs. "FSM for SSF/SRF interface" and "FSM for Specialized Resource" reflect the Enhanced SRF related processing as well as existing CS-1 SRF processing (refer to 12.5.1.3 and 12.5.1.4/Q.1228).

7.1.6 Out-channel Call Unrelated User Interaction (OCUUI)

"Out-channel Call Unrelated User Interaction (OCUUI)" IN CS-2 key function enables a service logic instance in the SCF to communicate with a user outside the context of a call (e.g. a service logic in the SCF communicates with a user who sends/receives information over out-channel signalling interface using the Q.932 protocol when no basic call processing is involved). OCUUI is required for services/service features such as terminal registration, out-channel based UPT location registration and message waiting indication, etc. For this capability, functional elements and service processing mechanism different from those for call related user interaction are specified. IN CS-2 specifications supporting this function are summarized in the following:

GFP specification items

- In the GFP, BCUP (Basic Call Unrelated Process) specialized SIB is newly introduced for the basic call unrelated processing and description for the USER INTERACTION SIB is enhanced for the SCF-User communication without the context of the call. "User Interaction Session Open", "User Interaction Session Close", "User Interaction Play" and "User Interaction Play and Collect" which are parts of USER INTERACTION SIB operations are used for OCUUI as well as call related user interaction (refer to 5.18/Q.1223 for USER INTERACTION SIB, and to 6.2/Q.1223 for BCUP specialized SIB).

DFP specification items

- In the IN CS-2 DFP, new FEs named "Call Unrelated Service Function" (CUSF) and "Service Control User Agent Function" (SCUAF) and the basic call unrelated processing model are defined to realize OCUUI function.
- The CUSF is a functional entity which performs call unrelated processing for the communication between a user and a service logic without a context of a call. It provides functions of:
  - association handling between the SCUAF;
  - detection of some events/triggers of call unrelated processing and reporting them to the SCF;
  - modification of call unrelated processing according to the SCF operations; and
  - support of out-channel user interaction.

  (Refer to 3.3.8/Q.1224).
- The SCUAF is one of user agent functions and it enables a user to access CUSF over a signalling interface. The relationship between the CUSF and the SCUAF is not the subject of IN standardization activities (refer to 7.1/Q.1221 and 3.3.9/Q.1224).
- The CUSF maintains the "Basic Call Unrelated State Model (BCUSM)" which models some aspects of the basic call unrelated processing in the CUSF such as association setup/release over the signalling channel or ROSE APDU reception but does not model the processing depending on the contents of APDU. The BCUSM is expressed as a combination of Points In Activation (PIAs) and Detection Points (DPs), where a PIA indicates association handling status and a DP indicates an event for association setup/release request or detection of a ROSE APDU reception. IN CS-2 BCUSM has three DPs and three PIAs, and three DP criteria are identified. The BCUSM defined in Recommendation Q.1224 is not a general state model of the basic call unrelated processing. The BCUSM may be enhanced or modified in future Capability Sets.
- The basic idea of BCUSM and service control mechanism for OCUUI is similar to the BCSM and the call related service control mechanism. The BCUSM hides the detail of call unrelated processing in the CUSF and is used for the detection of some triggers and events during the call unrelated processing and then the CUSF reports them to the SCF. In response to the DP notification from the CUSF, the service logic in the SCF sends back the operations to influence the call unrelated processing at the CUSF.
- The internal functional structure of the CUSF is similar to the SSF/CCF functional structure (refer to 8.1/Q.1224).
- There is some implication concerning relationships between the CUSF and the SSF/CCF but these relationships are not the subject of IN CS-2 specifications (refer to 8.3/Q.1224).
- Information Flows, Information Elements and corresponding operations newly defined for this function in IN CS-2 are listed in Tables 7-4 and 7-5 (refer to 12.7/Q.1224).

Table 7-4/Q.1229 – New IFs/IEs for OCUUI (from CUSF to SCF)

IF	IEs	Note
Activation Received And Authorized (12.7.2.1/Q.1224) <17.2/Q.1228>	Call ID (M), Service Address Information (M), Terminal Type (M), Calling Party Number (O), Component (O), Component Correlation ID (O), Component Type (O), Location Number (O)	This IF is issued by the CUSF for reporting the TDP event to the SCF that an association request (optionally with a request of an invocation of operation) has been received, and criteria for the Activation Received and Authorized DP were met.
Activity Test Response (12.7.2.3/Q.1224)	Call ID (M)	This IF is the response to the Activity Test For CUSF IF. At the operation level, The Return Result of ActivityTest operation corresponds to this IF.
Association Release Requested (12.7.2.4/Q.1224) <17.10/Q.1228>	Call ID (M), Service Address Information (M), Terminal Type (M), Calling Party Number (O), Component (O), Component Correlation ID (O), Component Type (O), Location Number (O)	This IF is issued by the CUSF for reporting the TDP/EDP event to the SCF that a request of association release with optionally an operation invocation request or an response/error has been received, and criteria for the Association release Requested DP were met.
Component Received (12.7.2.5/Q.1224) <17.29/Q.1228>	Call ID (M), Service Address Information (M), Terminal Type (M), Component Correlation ID (M), Component (O), Component Type (O), Calling Party Number (O), Location Number (O)	This IF is issued by the CUSF for reporting the TDP/EDP event to the SCF that an operation invocation request or an response/error has been received, and criteria for the Component Received DP were met.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Table 7-5/Q.1229 – New IFs/IEs for OCUUI (from SCF to CUSF)

IF	IEs	Note
Activity Test (12.7.2.2/Q.1224) <17.3/Q.1228>	Call ID (M)	This IF is used to check for the continued existence of a relationship between the SCF and CUSF. If the relationship is still in existence, then the CUSF will respond with Activity Test For CUSF Response. If no reply is received, then the SCF will assume that the CUSF has failed in some way and will take the appropriate action.
Initiate Association (12.7.2.6/Q.1224) <17.59/Q.1228>	Call ID (M), Called Party Number (M)	This information flow is used to allow the SCF to initiate a call unrelated association with the user.
Release Association (12.7.2.8/Q.1224) <17.83/Q.1228>	Call ID (M), Cause (M)	This IF is issued by the SCF for requesting the CUSF to release the logical connection.
Request Report BCUSM Event (12.7.2.7/Q.1224) <17.94/Q.1228>	Call ID (M), BCUSM Event List (M), Component Type (M), Component Correlation ID (O), Monitor Duration (O)	This IF is issued by the SCF for requesting the CUSF to report an EDP event to the SCF. The EDP event may be selectively reported by the CUSF with criteria for the DPs specified by this information flow, such as invoke, return result, etc., but this capability is optional.
Send Component (12.7.2.9/Q.1224) <17.112/Q.1228>	Call ID (M), Component Correlation ID (M), Message (M), Component (O), Monitor Duration (O), Component Type (O), Location Number (O)	This IF is issued by the SCF for requesting the CUSF to send a specified component to the SCUAF with a specified message. If the invocation from a network side takes place, the CUSF establishes a logical connection for a user with Called Party Number.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Protocol specification items

- The CUSF can be located at the local exchange level only in IN CS-2 (not at the transit exchange level).
- The CUSF supports only connection-oriented communication between a user and the network; connection-less communication is out of the scope of IN CS-2.
- The CUSF can be located at a node other than the SSP (e.g. SN). A new physical node named "Call Unrelated Service Point (CUSP)" is introduced in IN CS-2. The CUSP contains the CUSF and the CCF (refer to clauses 3 and 5/Q.1225).

- At the high level, CUSF internal structure for protocol processing and the CUSF Finite State Model (CUSF FSM) are specified for the IN CS-2 OCUUI implementation (refer to clause 15/Q.1228).

7.1.7 Out-Channel Call Related User Interaction (OCCRUI)

"Out-Channel Call Related User Interaction (OCCRUI)" provides the IN CS-2 compliant network the ability to transmit information between a user and a service logic instance within the context of a call on the out-channel signalling access. IN CS-2 specifications supporting this function are summarized in the following:

GFP specification items

- In the GFP, description for the USER INTERACTION SIB is enhanced for the Service logic/User communication within the context of the call using out-channel signalling. "User Interaction Session Open", "User Interaction Session Close", "User Interaction Play" and "User Interaction Play and Collect", which are parts of USER INTERACTION SIB operations, are used for OCCRUI as well as OCUUI (refer to 5.18/Q.1223).

DFP specification items

- SSF/CCF must realize signalling interworking between basic call signalling and INAP for OCCRUI processing (e.g. interworking between DSS1 functional protocol and INAP or interworking between ISUP and INAP) to transmit necessary information between a service logic and an ISDN user. This signalling interworking function for the OCCRUI capability can be located at a local exchange level or at a transit exchange level.
- The USI information element is introduced to carry information from an ISDN user to a service logic instance and from a service logic instance to an ISDN user. The SSF/CCF only transmits this IE transparently and does not touch the contents (refer to 4.2.9/Q.1224).
- A mechanism is specified (refer to 4.2.9/Q.1224) so that an SSF/CCF identifies the appropriate receiver of the information from an SCF/an ISDN user.
- Information Flows, Information Elements and corresponding operations newly defined for this function in IN CS-2 are listed in Tables 7-6 and 7-7 (refer to 12.4/Q.1224).

Table 7-6/Q.1229 – New IFs/IEs for OCCRUI (from SSF to SCF)

IF	IEs	Note
Event Report Facility (12.4.3.32/Q.1224) <17.50/Q.1228>	Call ID (M), Component (O), Component Correlation ID (O), Component Type (O), Leg ID (O)	This IF is issued by the SSF to report to the SCF that FACILITY IE received within appropriate DSS1 message. This flow is issued by SSF during the BCSM suspended at a Detection Point, when the SCF previously requested the event with the Request Report Facility Event IF.
Report UTSI (12.4.3.52/Q.1224) <17.87/Q.1228>	Call ID (M), Leg ID (M), USI Information (M), USI Service Indicator (M)	This IF is the response to the Request Report UTSI IF, when the monitoring has been previously requested. This IF is sent if a User to Service Information (UTSI) IE was received and the UTSI IE meets the conditions which were requested by the Request Report UTSI IF before.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Table 7-7/Q.1229 – New IFs/IEs for OCCRUI (from SCF to SSF)

IF	IEs	Note
Request Report Facility Event (12.4.3.55/Q.1224) <17.95/Q.1228>	Call ID (M), Component Correlation ID (O), Component Type (O), Leg ID (O), Monitor Duration (O)	This flow is issued by SCF for requesting SSF to report the event of FACILITY IE reception to SCF.
Request Report UTSI (12.4.3.56/Q.1224) <17.96/Q.1228>	Call ID (M), Leg ID (M), USI Monitor Mode (M), USI Service Indicator (M)	This IF is issued by the SCF to request the SSF to monitor for a User to Service Information (UTSI) information element. A notification is sent back to the SCF when the UTSI IE is detected by the SSF.
Send Facility Information (12.4.3.63/Q.1224) <17.113/Q.1228>	Call ID (M), Call Processing Correlation ID (O), Component (O), Component Correlation ID (O), Component Type (O), Leg ID (O)	This flow is issued by SCF for requesting SSF to send FACILITY IE to call party. It should also support FACILITY IE delivery within call establishing messages or Facility message.

Table 7-7/Q.1229 – New IFs/IEs for OCCRUI (from SCF to SSF) *(concluded)*

IF	IEs	Note
Send STUI (12.4.3.64/Q.1224) <17.114/Q.1228>	Call ID (M), Leg ID (M), USI Information (M), USI Service Indicator (M)	This IF is issued by the SCF to send a Service to User Information (STUI) information element to an user.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Protocol specification items

- The application entity procedure for the SSF is enhanced to include the OCCRUI FSM (Out-Channel Call Related User Interaction FSM) for this network aspect (refer to 11.8/Q.1228).

7.1.8 Service/Feature Interaction (Service Processing)

In the IN CS-2 time frame, only IN and non-IN service interaction is considered through all the planes of INCM. Regarding the service interaction between IN services is considered up to the DFP level (protocol specification will be considered in future capability sets).

GFP specification items

- Nothing identified.

DFP specification items

- The following three types of feature interaction are within the scope of IN CS-2 (2.11/Q.1224):
  - Case A: IN based to switch based: ServiceInteractionIndicator mechanism is used so that an IN service logic can allow/deny or modify switch-based service logic execution via call related signalling.
  - Case B: switch based to IN based: This case may be treated in the same way as Case C.
  - Case C: IN based to IN based: Two different approaches are identified. The first approach is based on the ServiceCompatibilityIndication parameter which is used during the triggering phase at the SSF to check the compatibility between two service logics. The compatibility checks can allow the triggering of a second TDP-R while there already exists an SSF-SCF control relationship. The second approach is based on the exchange of information between the two involved SCFs, but this approach is not specified in IN CS-2 and will be provided in IN CS-3 or later IN capability sets.

Protocol specification items

- Apart from their role during the triggering process in the SSF, the INServiceCompatibilityIndication and INServiceCompatibilityResponse parameters can be simply used to convey the list of services/service features subsequently invoked in the call.
- Feature Interaction may have to be managed between Single Point of Control services in a single SSF and between multiple SSFs, where as processing of the DP rules independently

ensures single point of control within many single call segments. Within a single SSF this may be achieved by static management procedures as in IN CS-1.

- Definition of the interaction between more than one SCFs for Feature Interaction Management is outside the scope of IN CS-2. Interaction between SSFs for Feature Interaction Management is included in IN CS-2 for those cases possible by ISUP parameter negotiation.

7.1.9 Internetworking between IN structured networks

This is one of the major functional requirements studied for IN CS-2. This function requires an extension of interfaces between FEs physically located in different networks so that these networks can cooperate together to provide a service (i.e. new information flows and information elements in IN DFP are required). Three relationships, SCF-SCF, SCF-SDF and SDF-SDF, are identified for internetworking in IN CS-2. IN CS-2 specifications supporting this function are summarized in the following:

GFP specification items

- For the SCF-SCF relationship, INITIATE SERVICE PROCESS SIB, END SIB and MESSAGE HANDLER SIB are introduced to handle parallel service processing. SIB operations of these SIBs, "Initiate Service Process", "End", "Send Message" and "Receive Message" are defined for this purpose (refer to 5.6, 5.7 and 5.10/Q.1223).
- AUTHENTICATE, LOG CALL INFORMATION, SCREEN, SERVICE DATA MANAGEMENT and TRANSLATE SIBs are enhanced to cover not only SCF-SDF relationship but also SDF-SDF relationship (refer to 5.2, 5.9, 5.12, 5.13 and 5.17/ Q.1223).

DFP specification items

- In IN CS-2, SCF-SCF relationship supports call related internetworking and SCF-SDF and SDF-SDF relationships support both call related and call unrelated-related internetworking. The latter case may be mostly used for terminal or personal mobility services/service features such as registration, authentication encryption and handover procedures. These relationships can be applied for both intranetworking case and internetworking case (refer to 3.4/Q.1224).

SCF-SCF relationship

- SCF-SCF relationship is used when a service logic instance in one SCF requires interactions with a service logic instance in another SCF (i.e. distribution of service logic). This means that the first SCF (controlling SCF) asks the second SCF (supporting SCF) to perform some action and the result of the action is returned to the first SCF. In another words, these service logic instances cooperate together in order to perform a required service (e.g. Customized Call Routing service). This is achieved by coordination, synchronization and security mechanisms embedded in the SCFs (refer to 3.4.2/Q.1224).
- Internetworking manager functional component is introduced to the SCF in order to support internetworking (refer to 6.2.2.5/Q.1224).
- Chaining and referral mechanisms are supported for the SCF-SCF relationship. The former is used for the case where the supporting SCF can not handle the request and transfer the request to the other SCF. The latter is used for the case where the supporting SCF can not handle the request and return to the controlling SCF an address information of an alternative SCF to which the request should be forwarded.
- Information Flows, Information Elements and corresponding operations defined for SCF-SCF relationship are listed in Tables 7-8 and 7-9 (refer to 12.6/Q.1224). Stage 2 descriptions of INITIATE SERVICE PROCESS and MESSAGE HANDLER SIBs will be helpful to understand relations among those information flows (refer to 11.2.7/Q.1224 and

11.2.10/Q.1224 respectively). Note that these IFs/IEs are not only for the internetworking but also for the intranetworking.

- For the SCF-SCF relationship, the same information flows and information elements as in the normal interworking case are used for the chaining case.

Table 7-8/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the controlling SCF to the supporting SCF)

IF	IEs	Note
Activity Test (12.6.2.1/Q.1224) <17.3/Q.1228>	Service Processing ID (M)	This IF is used to check for the continued existence of a relationship between the controlling SCF and the supporting SCF. If the relationship still needs to exist, then the SCF will respond with the Activity Test Result IF. If no reply is received, then the SCF issuing this IF will assume that the SCF has failed in some way and will take the appropriate action. No distinction of controlling/supporting side regarding this IF.
Additional Information Result (12.6.2.3/Q.1224)	Information (M), Service Processing ID (M), Security Information (O)	This IF sends back additional information to the supporting SCF that has requested it in order to assist the controlling SCF. It can also send back an indication that a user interaction has failed and that the user information could not be collected from the user. Result of ProvideUserInformation operation corresponds to this IF.
Confirmed Notification Provided (12.6.2.1/Q.1224) <17.30/Q.1228>, <17.17/Q.1228>	Request Confirmation (M), SCF Notification (M), Service Processing ID (M), Security Information (O)	This IF informs the service logic in the supporting SCF of service processing related information from the controlling SCF. The conditions of notification can either be requested to the controlling SCF by the reception of a Request Notification IF from the supporting SCF or be pre-arranged as part of the agreement between the two SCFs. The confirmation of this IF should be sent back using Notification Provided Confirmation IF. The MAKE CONFIRM syntax is applied to the NotificationProvided operation. Two operations, ConfirmedNotificationProvided and ChainedConfirmedNotificationProvided, correspond to this IF (for normal case and chaining case).

Table 7-8/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the controlling SCF to the supporting SCF) *(continued)*

IF	IEs	Note
Confirmed Report Charging Information (12.6.2.1/Q.1224) <17.31/Q.1228>, <17.30/Q.1228>	Calling Party Number (M), Service Processing ID (M), Request Confirmation (M), Account Number (O), Call Record (O), Called Party Number (O), Remaining User Credit (O), Security Information (O), Unique Call ID (O)	This IF is issued by the controlling SCF to the supporting SCF to provide it with the charging-related information which will be used as the charge record for a call in the controlling SCF. This IF may either be the response to the previously received Establish Charging Record IF or can be sent without having received the Establish Charging Record IF, in the pre-arranged case. In any case a Handling Information Request has been sent. The confirmation of this IF should be sent back using Report Charging Information Confirmation IF. The MAKE CONFIRM syntax is applied to the ReportChargingInformation operation. Two operations, ConfirmedReportChargingInformation and ChainedReportChargingInformation, correspond to this IF (for normal case and chaining case).
Handling Information Request (12.6.2.8/Q.1224) <17.55/Q.1228>, <17.19/Q.1228>	Service Processing ID (M), Active Supplementary Services (O), Bearer Capability (O), Called Party Number (O), Calling Party Number (O), Calling Party Business Group ID (O), Calling Party's Category (O), Cause Of Last Call Failure (O), Dialled Digits (O), High Layer Compatibility (O) Input Information (O), Invoked Supplementary Services (O), Location Number (O), Number Of Call Attempts (O), Original Called Party ID (O), Redirecting Party ID (O), Redirection Information (O), Requested Type (O), Security Information (O), User Interaction Mode (O)	This IF is issued by the controlling SCF for requesting a call processing information to the other SCF, or for requesting the other SCF to perform the predefined actions. The requested information is returned in a Handling Information Result IF. The presence of the parameters in the IF is dependent of the Service Logic type exchanged in the SCF Bind Request IF. This IF will not be sent empty. Two operations, HandlingInformationRequest and ChainedHandlingInformationRequest, corresponds to this IF (for normal case and chaining case).

Table 7-8/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the controlling SCF to the supporting SCF) *(continued)*

IF	IEs	Note
Network Capability Result (12.6.2.11/Q.1224)	Service Processing ID (M), Bearer Services (O), Security Information (O), Supplementary Services (O), Teleservices (O)	This IF is a response to the Network Capability Request IF. Result of NetworkCapability operation corresponds to this IF.
Notification Provided (12.6.2.12/Q.1224) <17.68/Q.1228>, <17.22/Q.1228>	Service Processing ID (M), SCF Notification (M), Security Information (O)	This IF informs the service logic in the supporting SCF of service processing related information from the controlling SCF. The conditions of notification can either be requested to the controlling SCF by the reception of a Request Notification IF from the supporting SCF or be pre-arranged as part of the agreement between the two SCFs. Two operations, NotificationProvided and ChainedNotificationProvided, correspond to this IF (for normal case and chaining case).
Report Charging Information (12.6.2.16/Q.1224) <17.86/Q.1228>, <17.23/Q.1228>	Service Processing ID (M), Calling Party Number (M), Account Number (O), Call Record (O), Called Party Number (O), Remaining User Credit (O), Security Information (O), Unique Call ID (O)	This IF is issued to provide the charging-related information which will be used as the charge record for a call in the controlling SCF. This IF may either be the response to the previously received Establish Charging Record IF or can be sent without having received the Establish Charging Record IF, in the pre-arranged case. In either case a Handling Information Request has been sent. There is no confirmation of this IF. Two operations, ReportChargingInformation and ChainedReportChargingInformation, corresponds to this IF (for normal case and chaining case).
SCF Bind Request (12.6.2.18/Q.1224) <17.100/Q.1228>, <17.101/Q.1228>	Agreement ID (M), Service Processing ID (M), SCF Address (O), Security Information (O)	This IF is used to establish relationship between two SCFs. This information flow is sent by a controlling SCF each time it needs to initiate communications with the supporting SCF and to ensure that the called entity has all facilities to operate on messages to be sent. SCFBind operation is used for both normal and chaining cases.

Table 7-8/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the controlling SCF to the supporting SCF) *(concluded)*

IF	IEs	Note
SCF Unbind Request (12.6.2.20/Q.1224) <17.102/Q.1228>, <17.103/Q.1228>	Service Processing ID (M)	This IF is used to request the termination of the active association with the supporting SCF. It can be sent only by the controlling SCF. SCFUnbind operation is used for both normal and chaining cases.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Table 7-9/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the supporting SCF to the controlling SCF)

IF	IEs	Note
Activity Test Result (12.6.2.2/Q.1224)	Service Processing ID (M)	Result of activityTest operation corresponds to this IF. No distinction of controlling/supporting side regarding this IF
Establish Charging Record (12.6.2.6/Q.1224) <17.46/Q.1228>, <17.18/Q.1228>	Service Processing ID (M), Charging Parameters (O), Report Expected (O), Security Information (O), User Credit (O)	This IF is issued to give the controlling SCF charging related information needed for the call to proceed, e.g. charging rate information and maximum allowed credit. When the call instance is terminated, a response is returned to the supporting SCF. This IF is a way to enable both SCFs to manage the charging information without pre-defined manner (this means the charging-related information will vary call by call even if the same service/service features are invoked in the controlling SCF. Two operations, EstablishChargingRecord and ChainedEstablishChargingRecord, corresponds to this IF (for normal case and chaining case).

Table 7-9/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the supporting SCF to the controlling SCF) *(continued)*

IF	IEs	Note
Handling Information Referral (12.6.2.7/Q.1224)	Service Processing ID (M), Referral Information (M)	This IF is the response to the Handling Information IF in the case where the supporting SCF does not contain the data involved in the request, and is used to provide the controlling SCF with the information required to redirect the query to another supporting SCF. ERROR part of HandlingInformationRequest operation corresponds to this IF.
Handling Information Result (12.6.2.9/Q.1224) <17.56/Q.1228>, <17.20/Q.1228>	Service Processing ID (M), Calling Party Number (O), Calling Party's Category (O), Carrier (O), High Layer Compatibility (O), Language ID (O), Original Called Party ID (O), Output Information (O), Redirecting Party ID (O), Redirection Information (O), Routing Address (O), Security Information (O), Supplementary Services (O)	The requested information using Handling Information Request IF is returned. Two operations, HandlingInformationResult and ChainedHandlingInformationResult, correspond to this IF (for normal case and chaining case).
Network Capability Request (12.6.2.10/Q.1224) <17.66/Q.1228>, <17.21/Q.1228>	Service Processing ID (M), Bearer Services (O), Security Information (O), Supplementary Services (O), Teleservices (O)	This IF enables the supporting SCF to request the type of services that can be fulfilled by the controlling SCF, if not already specified by the agreement. It must be preceded by a Handling Information Request IF. The requested information is returned in the Network Capability Result. It gives the level of service that can be expected from the controlling SCF. This type of information can be used to build the response to the initial Handling Information Request. Two operations, NetworkCapability and ChainedNetworkCapability, correspond to this IF (for normal case and chaining case).

Table 7-9/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the supporting SCF to the controlling SCF) *(continued)*

IF	IEs	Note
Notification Provided Confirmation (12.6.2.13/Q.1224)	Service Processing ID (M), Security Information (O)	This IF is issued to confirm the Confirmed Notification Provided IF reception. Result of ConfirmedNotificationProvided operation corresponds to this IF.
Provide User Information (12.6.2.14/Q.1224) <17.81/Q.1228>, <17.24/Q.1228>	Constraints (M), Information To Send (M), Number Of Allowed Retries (M), Service Processing ID (M), Type Of Requested Info (M), Actions (O), Error Info (O), Language ID (O), Security Information (O)	This IF is used by the supporting SCF to request additional information from the controlling SCF. This IF is initiated when the supporting SCF receives a Handling Information Request IF from the controlling SCF and it detects that additional information is needed from the calling user/controlling SCF in order for the call to proceed. The controlling SCF returns the information to the supporting SCF by the Additional Information Result IF. The supporting SCF may invoke multiple Provide User Information IFs. Two operations, ProvideUserInformation and ChainedProvideUserInformation, corresponds to this IF (for normal case and chaining case).
Report Charging Information Confirmation (12.6.2.16/Q.1224)	Service Processing ID (M), Security Information (O)	This IF is issued to confirm the Confirmed Report Charging Information IF reception. Result of ConfirmedReportChargingInformation operation corresponds to this IF.
Request Notification (12.6.2.17/Q.1224) <17.91/Q.1228>, <17.25/Q.1228>	Requested Notifications (M), Service Processing ID (M), Security Information (O)	This IF is issued to request notifications of the service processing related information from the controlling SCF. Two operations, RequestNotification and ChainedRequestNotification, correspond to this IF (for normal case and chaining case).

Table 7-9/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from the supporting SCF to the controlling SCF) *(concluded)*

IF	IEs	Note
SCF Bind Result (12.6.2.19/Q.1224)	Service Processing ID (M), Security Information (O), Supporting SCF Address (O)	This IF is used by the supporting SCF to respond to the request for an association from the controlling SCF. Before the supporting SCF sends the positive SCF Bind Result, it shall not accept any other messages from the controlling side for this association. Neither shall it send any messages for this association to the controlling SCF other than SCF Bind Result. Result of SCFBind operation corresponds to this IF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

SCF-SDF relationship

- Referral mechanism is supported for the SCF-SDF relationship.
- Information Flows, Information Elements and corresponding operations newly defined for SCF-SDF relationship in IN CS-2 are listed in Tables 7-10 and 7-11 (see 12.8/Q.1224). Note that these IFs/IEs are not only for the internetworking but also for the intranetworking.
- IN CS-2 introduces generic security framework for secured access of a service user to a FE when internetworking activity is involved (refer to 7.1.11). End Authenticated Relationship information flow is introduced according to this framework in addition to the existing Authenticate and Authenticate Result information flows.
- "Execute" IF/operation is introduced for efficient access to data in the SDF from the SCF. The background of the introduction of this IF/operation is described in the context of "Entry Method" in 7.2.3.4.

Table 7-10/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from SCF to SDF)

IF	IEs	Note
End Authenticated Relationship (12.8.2.6/Q.1224) <17.38/Q.1228>	Authorized Relationship ID (M)	This IF is issued by the SCF to end an authenticated relationship between the SCF and the SDF on behalf of the end user. IN-directoryUnbind operation corresponds to this IF.
Execute (12.8.2.7/Q.1224) <17.51/Q.1228>	Authorized Relationship ID (M), Execute Identifier (M), Object (M), Specific Input Value (M), Input Attributes (O)	This IF is used to request the SDF to perform the data access script associated with a particular item of data held in the DIT in the SDF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Table 7-11/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from SDF to SCF)

IF	IEs	Note
Execute Result (12.8.2.9/Q.1224)	Authorized Relationship ID (M), Specific Output Value (M), Output Attributes (O)	This IF is the response to the Execute IF. RESULT part of Execute operation corresponds to this IF
Add Entry Referral (12.8.2.2/Q.1224)	Authorized Relationship ID (M), Referral Information (M)	This IF is the response to the Add Entry IF in the case where the SDF does not contain the data involved in the request, and is used to provide the SCF with the information required to redirect the query to another SDF. ERROR part of the AddEntry operation corresponds to this IF.
Execute Referral (12.8.2.8/Q.1224)	Authorized Relationship ID (M), Referral Information (M)	This IF is the response to the Execute IF in the case where the SDF does not contain the data involved in the request, and is used to provide the SCF with the information required to redirect the query to another SDF. ERROR part of the Execute operation corresponds to this IF.

Table 7-11/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from SDF to SCF) *(concluded)*

IF	IEs	Note
Modify Entry Referral (12.8.2.11/Q.1224)	Authorized Relationship ID (M), Referral Information (M)	This IF is the response to the Modify Entry IF in the case where the SDF does not contain the data involved in the request, and is used to provide the SCF with the information required to redirect the query to another SDF. ERROR part of the ModifyEntry operation corresponds to this IF.
Remove Entry Referral (12.8.2.14/Q.1224)	Authorized Relationship ID (M), Referral Information (M)	This IF is the response to the Remove Entry IF in the case where the SDF does not contain the data involved in the request, and is used to provide the SCF with the information required to redirect the query to another SDF. ERROR part of the RemoveEntry operation corresponds to this IF.
Search Referral (12.8.2.17/Q.1224)	Authorized Relationship ID (M), Referral Information (M)	This IF is the response to the Search IF in the case where the SDF does not contain the data involved in the request, and is used to provide the SCF with the information required to redirect the query to another SDF. ERROR part of the Search operation corresponds to this IF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

SDF-SDF relationship

- The SDF in IN CS-2 provides data distribution transparency, data copy between different SDFs and security functionalities which are used when internetworking activity takes place (refer to 3.3/Q.1224).
- Copying the data through the SDF-SDF relationship is referred to as "Shadowing". The SDF which supplies copy data to the other SDF is referred to as the "supplier" and the SDF which receives copy data is referred to as the "consumer". Two different shadowing cases, "Supplier-initiated shadow updates" and "Consumer-initiated shadow updates" are described in 7.2.3.7.
- Chaining and referral mechanisms are supported for the SDF-SDF relationship. The former is used for the case where the requested SDF does not have the requested data and transfers the request to the other SDF; this mechanism enables data distribution transparency. The latter is used for the case where the requested SDF does not have the requested data and returns an address information of an alternative SDF to the requesting SDF for the redirection of the request.

- Information Flows, Information Elements and corresponding operations defined for SDF-SDF relationship in IN CS-2 are listed in Table 7-12 (see 12.9/Q.1224). Note that these IFs/IEs are not only for the internetworking but also for the intranetworking.

Table 7-12/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from SDF to SDF)

IF	IEs	Note
Authenticate (12.9.2.1/Q.1224) <17.42/Q.1228>, <17.42/Q.1228>	Authentication Information (M), Authorized Relationship ID (M)	DSABind and DSAShadowBind operations correspond to this IF (for normal case and shadowing case).
Authenticate Result (12.9.2.2/Q.1224)	Authentication Information (M), Authorized Relationship ID (M)	Results of DSABind and IN-DSAShadowBind operations correspond to this IF.
Chaining Request (12.9.2.3/Q.1224)	Authorized Relationship ID (M), Chained Argument (M), Security Parameters (M)	Chained{OPERATION}[a)] corresponds to this IF.
Chaining Result (12.9.2.4/Q.1224)	Authorized Relationship ID (M), Chained Result (M), Security Parameters (M)	Result of Chained{OPERATION}[a)] corresponds to this IF.
Copy Request (12.9.2.5/Q.1224) <17.36/Q.1228>, <17.97/Q.1228>	Authorized Relationship ID (M), Maintained Part (M), Master (M), Replication Area (M), Update Mode (M), Update Strategy (M)	CoordinateShadowUpdate and RequestShadowUpdate operations correspond to this IF[b)].
Copy Result (12.9.2.6/Q.1224)	Authorized Relationship ID (M), Replicated Data (M)	Result of CoordinateShadowUpdate operation or Result of RequestShadowUpdate operation corresponds to this IF.
End Authenticated Relationship (12.9.2.7/Q.1224) <17.58/Q.1228>, <17.43/Q.1228>	Authorized Relationship ID (M)	This IF is issued by the SDF to end an authenticated relationship between two SDFs. IN-DSAUnbind and IN-DSAShadowUnbind correspond to this IF (for normal case and shadowing case).
Update Copy (12.9.2.8/Q.1224) <17.127/Q.1228>	Authorized Relationship ID (M), Refreshed Information (M)	This IF is used to maintain a copy contained in the SDF to which a copy was originally provided because the selected update mode indicates that an update of the copy should be sent (e.g. modification of the copy in the responding network). UpdateShadow operation corresponds to this IF.
Update Copy Result (12.9.2.9/Q.1224)	Authorized Relationship ID (M)	Result of UpdateShadow operation corresponds to this IF.

Table 7-12/Q.1229 – New IFs/IEs for internetworking between IN structured networks (from SDF to SDF) *(concluded)*

NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF.

NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column.

NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.

a) Chained{OPERATION} is abbreviation for the list of the following operations:

- ChainedAddEntry <17.17/Q.1228>
- ChainedExecute <17.19/Q.1228>
- ChainedModifyEntry <17.21/Q.1228>
- ChainedRemoveEntry <17.25/Q.1228>
- ChainedSearch <17.26/Q.1228>

b) Sending the data from the supplier to the consumer is done by UpdateShadow operation. But before sending the data, one of the two procedures in the following must be preceded:

- the consumer indicates the supplier request for copying (or updating) the shadow data;
- the supplier indicates the consumer shadowing agreement for which it intends to send updates.

RequestShadowUpdate operation is used for the first case, sent from the consumer to the supplier ("Consumer initiated shadow updates" case).

CoordinateShadowUpdate operation is used for the second case, sent from the supplier to the consumer ("Supplier initiated shadow updates" case). (refer to 17.36.1/Q.1228 and to 7.2.3.7).

Protocol specification items

- For the SCF-SDF and SDF-SDF relationships, a subset of X.500-series Recommendation, Directory Service protocol specifications are utilized in IN CS-2. Restrictions and assumptions on utilizing X.500 series are provided in clauses 7 and 8/Q.1228, for each relationship.
- In IN CS-2, SCF Call State Model (SCSM) has several kinds of sub-state-model for each relationship with interworking FE (refer to 12.3/Q.1228).

  SCSM-Sup (state transition model for the supporting SCF), SCSM-Con (state transition model for the controlling SCF), SCSM-ChI (state transition model for the chaining initiation SCF) and SCSM-ChT (state transition model for the chaining termination SCF) are defined for the SCF-SCF relationship (refer to 12.5.3/Q.1228).

  SCSM-SDF (state transition model for interworking with SDF) is defined for the SCF-SDF relationship (refer to 12.5.2/Q.1228).
- In IN CS-2, SDF Call State Model (SDSM) has several kinds of sub-state-model for each relationship with interworking FE (refer to 14.3/Q.1228).

  SDSM-ShSSi (state transition model for the supplier SDF when shadowing is initiated by the supplier), SDSM-ShCSi (state transition model for the consumer SDF when shadowing is initiated by the supplier), SDSM-ShSCi (state transition model for the supplier SDF when shadowing is initiated by the consumer) and SDSM-ShCCi (state transition model for the consumer SDF when shadowing is initiated by the consumer) are defined for shadowing process on the SDF-SDF relationship (refer to 14.4.2.1/Q.1228).

SDSM-ChI (state transition model for the chaining initiation SDF) and SDSM-ChT (state transition model for the chaining termination SDF) are defined for chaining process on the SDF-SDF relationship (refer to 14.4.2.2/Q.1228).

SDSM-SCF (state transition model for SCF interworking) is defined for the SDF-SCF relationship (refer to 14.4.1/Q.1228).

- Relations between SCF-SCF, SCF-SDF and SDF-SDF related protocol specifications and TC specifications are provided in 18.1.5, 18.1.6 and 18.1.7/Q.1228 respectively (e.g. mapping of some operations to TC dialogue primitives are provided).

7.1.10 Internetworking with non-IN structured networks

GFP specification items

- SCF-SCF interface related SIBs are applicable (refer to the previous subclause).

DFP specification items

- A new FE called IAF (Intelligent Access Function) residing in an entity of a non-IN structured network is identified for the communication with an SCF in an IN structured network (refer to 3.3.7/Q.1224).
- Two different relationships between the SCF and the IAF are necessary for two different cases from the security, charging and reliability requirements point of view (refer to 3.4.3/Q.1224).
  - Case A: when the IAF belongs to another network;
  - Case B: when the IAF belongs to a customer (e.g. private networks, PABXs and terminals, etc.).
- Information flows and information elements between the SCF and the IAF are same as those for SCF-SCF interface (refer to 12.6/Q.1224).

Protocol specification items

- INAP operations for SCF-IAF interface are same as those for SCF-SCF interface.

7.1.11 Security

General requirements for secure systems are described in 7.2.10/Q.1221. Security aspects in the IN CS-2 scope is focused on the "Service User Authentication" which provides secure access by a user to the functions of a FE. This capability is specified for the internetworking relationship, i.e. SCF-SDF, SCF-SCF and SDF-SDF.

GFP specification items

- AUTHENTICATE SIB is enhanced for this function. This SIB is related to the SCF-SDF and SDF-SDF security function (refer to 5.2/Q.1223).
- INITIATE SERVICE PROCESS and END SIBs are related to the SCF-SCF security functions (refer to 5.6 and 5.7/Q.1223).

DFP specification items

- The high-level description of the "service user authentication" mechanism and generic security information flows including chaining and referral cases are specified in order to protect FEs from illegal access from an unauthorized user (refer to 11.1.6/Q.1224).
- SCF and SDF contain Security Manager functional component to realize security function (refer to 6.2.7 and 7.2.4/Q.1224).

- Information Flows, Information elements and corresponding operations newly defined for this function in IN CS-2 are listed in Tables 7-13 to 7-16 (refer to 11.6.3, 12.6, 12.8 and 12.9/Q.1224).

Table 7-13/Q.1229 – New IFs/IEs for security (between two SCFs)

IF	IEs	Note
SCF Bind Request (12.6.2.18/Q.1224) <17.100/Q.1228>, <17.101/Q.1228>	Agreement ID (M), Service Processing ID (M), SCF Address (O), Security Information (O)	This IF is used to establish relationship between two SCFs. This information flow is sent by a controlling SCF each time it needs to initiate communications with another (supporting) SCF and to ensure that the called entity has all facilities to operate on messages to be sent. SCFBind operation corresponds to this IF.
SCF Bind Result (12.6.2.19/Q.1224)	Service Processing ID (M), Security Information (O), Supporting SCF Address (O)	This IF is used by the supporting SCF to respond to the request for an association from the controlling SCF. It is mapped to the Return Result of the SCFBind operation.
SCF Unbind Request (12.6.2.20/Q.1224) <17.102/Q.1228>, <17.103/Q.1228>	Service Processing ID (M)	This IF is used to request the termination of the active association with the supporting SCF. It can be sent only by the controlling SCF. SCFUnbind operation corresponds to this IF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Table 7-14/Q.1229 – New IFs/IEs for security (from SCF to SDF)

IF	IEs	Note
Authenticate (12.8.2.4/Q.1224)	Authentication Information (M), Authorized Relationship ID (M)	This IF is used to request the establishment of the authenticated relationship between the SCF and the SDF on behalf of the end user DirectoryBind operation corresponds to this IF.
End Authenticated Relationship (12.8.2.6/Q.1224) <17.38/Q.1228>	Authorized Relationship ID (M)	This IF is issued by the SCF to end an authenticated relationship between the SCF and the SDF on behalf of the end user. DirectoryUnbind operation corresponds to this IF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Table 7-15/Q.1229 – New IFs/IEs for security (from SDF to SCF)

IF	IEs	Note
Authenticate Result (12.8.2.5/Q.1224)	Authorized Relationship ID (M), Authentication Information (O)	This IF is used to confirm an establishment of an authenticated relationship between the SCF and the SDF on behalf of the end user. The Return Result of the DirectoryBind operation corresponds to this IF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If the name of the operation is different from that of the IF, it is shown in the last column.		

Table 7-16/Q.1229 – New IFs/IEs for Security (between two SDFs)

IF	IEs	Note
Authenticate (12.9.2.1/Q.1224) <17.42/Q.1228>	Authentication Information (M), Authorized Relationship ID (M)	This IF is used to have identification and authentication of two SDFs involved in an SDF-SDF relationship. This IF is prior to any IF on the SDF-SDF interface. It is used to enforce access control policy between databases. DSABind or DSAShadowBind operation corresponds to this IF.
Authenticate Result (12.9.2.2/Q.1224)	Authorized Relationship ID (M), Authentication Information (O)	This IF is used to confirm the result of authentication by the interacting SDF. The Return Result of the DSABind or DSAShadowBind operation corresponds to this IF.
End Authenticated Relationship (12.8.2.6/Q.1224) <17.43/Q.1228>	Authorized Relationship ID (M)	This IF is issued by the SDF to end an authenticated relationship between two SDFs. DSAUnbind and DSAShadowUnbind operations correspond to this IF.
NOTE 1 – The subclause number of Recommendation Q.1224 in the first column indicates the reference for the IF. NOTE 2 – If an operation corresponding to the IF exists, reference to Recommendation Q.1228 for the detailed procedure of the corresponding operation is provided in brackets in the first column. NOTE 3 – If there is no operation which has the same name as the IF, the last column shows how to map the IF to an IN CS-2 protocol element. If there is no such description in the last column, the name of the corresponding operation is the same as that of the IF.		

Protocol specification items

- Subclause 18.1.5.3.6/Q.1228 provides how to use the TC dialogue handling services for establishing/releasing the authentication relationship between SCFs.
- Subclause 18.1.6.3.6/Q.1228 provides how to use the TC dialogue handling services for establishing/releasing the authentication relationship between SCF and SDF.
- Subclause 18.1.7.3.6/Q.1228 provides how to use the TC dialogue handling services for establishing/releasing the authentication relationship between two SDFs.
- Generic IN CS-2 security mechanisms for the interfaces identified above are provided in clause 19/Q.1228. Requirements for interface security, necessary procedures and definitions of FSMs for processing the security operations are described. Appendix III/Q.1228 indicates some examples of Simple Public Key GSS API Mechanism (SPKM) algorithms.

7.1.12 Personal Mobility

This mobility support requires an extension of the IN model and architecture to guarantee a correct handling of the user profile and service processing, independently from the user access. This function depends on other network functionalities, either already provided in IN CS-1, or provided in IN CS-2.

7.1.13 Terminal Mobility

This function is studied and specified so that a part of a mobile network (e.g. IMT-2000: International Mobile Telecommunications-2000) is structured based on the IN architecture. I.3.3.2/Q.1221 lists IN CS-2 target Terminal Mobility services/service features. DFP architecture is

enhanced to support this function. Protocol specifications for this function are out of the scope of IN CS-2.

GFP specification items

No terminal mobility specific SIBs or GFP modelling elements are specified. SIBs and GFP modelling elements specified in Recommendation Q.1223 are used for this function.

DFP specification items

- New FEs shown in the following are introduced for the terminal mobility.
  - CRACF (Call related Radio Access Control Function): realizes terminal mobility specific call/bearer control functions such as handover or paging.
  - CURACF (Call Unrelated Radio Access Control Function): detects call unrelated events from a mobile terminal and notify it to an SCF. It also transfer information between a mobile terminal and the SCF.
  - RCF (Radio Control Function): maintains radio and fixed bearer to establish and release a communication path between a mobile terminal and the network.

  CCF and CCAF are extended to support the terminal mobility function and they are denoted as CCF+ and CCAF+.
  - CCF+: is enhanced from the CCF in order to interwork with CCAF+, CRACF and CURACF.
  - CCAF+: provides the user agent function to access CCF+, CRACF, CURACF and RCF.

  Definitions of these FEs and relationships identified for the terminal mobility are described in A.3/Q.1224. Example mapping scenarios of terminal mobility related FEs onto physical entities are provided in A.4/Q.1224.
- Appendix II/Q.1224 provides informative descriptions about the frameworks of call related processing at the CRACF and non-call related processing at the CURACF and also provides IFs/IEs for the SCF-SSF/CCF, SCF-CRACF and SCF-CURACF relationships.

7.1.14 IN-TMN

In IN CS-2, the Telecommunication Management Network (TMN) concept is introduced to define the framework for IN service and network management. Annex B/Q.1224 describes overview of the TMN concepts and provides some considerations on how the concepts are applied to the IN DFP architecture. The SMF functions could be mapped to more than one TMN layers. Examples of the mapping are shown in the Annex B/Q.1224. See also the next clause.

7.1.15 Service Management

IN CS-2 Recommendations specify a framework of IN service management functional architecture. Subclause 7.2.12/Q.1221 provides the IN CS-2 target scope in terms of service management aspects and required functionalities. Subclause A.4/Q.1221 indicates IN CS-2 target Service Management services/service features. Protocol specifications for service management are out of the scope of IN CS-2.

GFP specification items

- Appendix I/Q.1223, includes informative texts of management aspects at GFP. It shows a consideration on GFP modelling for the management activities and gives the following elements:
  - BSMP (Basic Service Management Process) is a specialized SIB which provides basic service management capability.
  - Management Process is a combination of SIBs or HLSIBs which performs management activity.

The approach for the modelling of management activities using these elements are similar to the approach for the modelling of the service processing activities using the Basic Call Process (BCP) and the Service Process.

DFP specification items

- Categories of IN CS-2 management functions of Service Management Function (SMF) are:
  - Service Deployment functions;
  - Service Provisioning functions;
  - Service Operation Control functions;
  - Billing functions; and
  - Service Monitoring functions.

  (Refer to 3.3.10/Q.1224).

  Considerations on mapping of these IN management functions to TMN architecture are shown in B.4/Q.1224.
- Relationships between the SMF and other FEs identified in IN CS-2 are:
  - SMF-SCF;
  - SMF-SDF;
  - SMF-SSF/CCF;
  - SMF-SRF;
  - SMF-SMAF;
  - SMF-SCEF;
  - SMF-SMF; and
  - SMF-CUSF.

  (Refer to 3.4/Q.1224).

  Information flows and information elements for these relationships are not specified in IN CS-2 Recommendations. Subclause B.6/Q.1224 describes some considerations on SMF-SMF internetworking aspect.
- The SMF consists of seven functional components: FE Access Manager, Security Access Manager, Configuration Manager, Fault Manager, Performance Manager, Testing Manager and Security Control Manager (refer to clause 9/Q.1224).
- A method for establishing management information models for IN FEs is shown in Annex C/Q.1224. The method adopts three steps:
  - decompose the FE to identify sub-entities to which management operations are applied;
  - clarify the FE management requirements to identify a set of management operations applied on the FE sub-entities; and
  - specify management information model by taking account of the output of the previous steps.

  The SSF is taken as an example to show the detailed procedure for each step of the proposed method. An example of the management information model of the SSF/CCF derived from the method is shown in the Appendix I/Q.1224.
- Some of the testing functions necessary for the SSF/CCF to check integrity of the IN service processing functions in the SSF/CCF and usage of these testing functions are provided. An end-to-end testing capability is also considered to check out the fault node by using some parameters dedicated for the testing purpose, where these parameters are passed around the nodes and collect the necessary information (refer to Annex D/Q.1224).

Protocol specification items

Out of the scope of IN CS-2.

7.1.16 Service Creation

IN CS-2 Recommendations specify few things about this aspect. Subclause 7.2.13/Q.1221 provides the IN CS-2 target scope of this aspect and required functionalities. Subclause A.5/Q.1221 indicates IN CS-2 target Service Creation services/service features. Subclause B.4.3/Q.1224 provides a consideration on decomposition of the SCEF into TMN logical layers.

7.1.17 GFP modelling and Service Independent Building Blocks for IN CS-2

IN CS-2 enhances GFP in various aspects. Enhancements regarding general aspects are described in Recommendation Q.1203 and IN CS-2 specific aspects are described in Recommendation Q.1223. This subclause provides an overview of IN CS-2 enhancements. The enhancements made include not only the SIBs modification but also the introduction of new concepts to GFP modelling.

7.1.17.1 GFP modelling

IN CS-2 enhances the description capability of GFP capabilities and GSL (Global Service Logic) by introducing new concepts and elements outlined in the followings (refer to clause 4/Q.1223).

- *Capability View and Service View*

  "Capability View" provides the capabilities of GFP within a single "Domain" by listing "SIBs" and "SIB operations". Within IN CS-1, SIBs are only means to describe GFP capabilities. IN CS-2 introduces new elements, "SIB operation", which has more precise level of granularity to describe capabilities of GFP. The set of SIBs and SIB operations expresses the capabilities of GFP and these also serve as basic elements to describe any GSL in GFP.

  "Service View" provides how IN CS-2 services or service features are realized. A service or service feature in GFP, namely a GSL, is expressed by a sequence of "High Level SIBs (HLSs)", "SIBs" and "SIB operations", while an IN CS-1 GSL is expressed by a sequence consisting of only SIBs.

  IN CS-2 largely enhances description capability of dynamic behaviour of GSLs (initiation, execution and termination of GSLs). This is done by "Service Process", where multiple service processes run concurrently and communicate with each other. By this enhancement, various services or service features having parallel processing activities can be described as GSLs.

- *Domain*

  A domain expresses an area where a "service process" can be executed and separated from another domain by boundary.

- *SIB operation*

  IN CS-2 introduced granularity into the SIB concept. A "SIB operation" is an atomic element of the SIB and a SIB generally consists of multiple SIB operations. SIB operations could form a GSL together with SIBs and/or HLSs. Three kinds of data – "Call Instance Data", "Service Support Data" and "Service Instance Data" – are used to define a SIB operation. SIB operations add more flexibility in describing GSLs to IN CS-1 GSL description capability.

- *High Level SIB*

  The High Level SIB (HLS) provides another level of granularity to the SIB concept. A HLS composed of SIB operations and other HLS and is considered as a reusable part of GSLs. A HLS hides detailed information about its internal logic and some of "Service Support Data"

which are local to the HLS. This characteristic will help a service designer to create GSLs easily.

- *Service Process*

  A service process represents a GSL instance which resides in a single domain. A service process can not exist over multiple domains but can initiate another process in a different domain and they can run concurrently. A service process can communicate with other processes through "Point Of Synchronization" (refer to 4.5.5 and 4.5.6/Q.1223 for communications between processes).

7.1.17.2 Service Independent Building Blocks (SIBs)

In IN CS-2, 21 SIBs are defined, the IN CS-1 list of SIBs being enhanced by seven extra SIBs shown in the following to support the list of targeted services and service features identified in Recommendation Q.1221 (Refer to clause 5/Q.1223 for each SIB definition):

- END: indicates the normal end of service process, or part of an service process in case of multiple threads. It also ends the authorization relationships between the service processes.
- INITIATE SERVICE PROCESS: causes the execution of the parallel service process to begin and it also establishes an authorization relationship for a user between the service processes.
- JOIN: attaches a call party or a group of call parties from the current call group into an indicated group within the same call.
- MESSAGE HANDLER: sends a message conveyed with Inter Process Data between a controlling and a supporting service process (refer to 4.1.2.3/Q.1223 for "controlling" and "supporting" service process).
- SPLIT: detaches a call party or group of call parties from the current call and attaches the indicated call parties in new initiated call or another existing call.
- Basic Call Unrelated Process (BCUP): is a specialized SIB which provides the call unrelated capabilities. BCUP is an independent process which communicates with CUUI (Call Un-related User Interaction) Service Processes. The concept is similar to that of "Basic Call Process" (refer to 6.2/Q.1223 for more details).

END, INITIATE SERVICE PROCESS and MESSAGE HANDLER SIBs are introduced to describe processing activities in GFP. JOIN and SPLIT SIBs are introduced to describe CPH activities in GFP and BCUP SIB is introduced to model interworking process between a basic call unrelated process and a OCUUI GSL.

The following SIBs have existed since IN CS-1 and re-defined in IN CS-2 by using IN CS-2 SIB definition method which adopts SIB operation concept (refer to 4.3.3/Q.1223 for the method and to clause 5/Q.1222 for the definition of each SIB). Some of these SIBs have enhanced capabilities to fulfill the IN CS-2 capability requirements:

- ALGORITHM: applies a mathematical algorithm to data to produce a data result.
- AUTHENTICATE: provides the functionality necessary to establish a relationship between service logic and service data based on a specific user identity. This identity is used by subsequent service data access operations to determine if the user identity has the necessary access privileges to perform the requested operations.

  This SIB has enhanced its capability so as to meet the IN CS-2 security requirements.
- CHARGE: determines special charging treatment for the call, where "special" refers to any charging in addition to that normally performed by the basic call process.

  In IN CS-2, this SIB provides the functionality for producing the data to be recorded physically.
- COMPARE: performs a comparison of an identifier against a specified reference value.

- DISTRIBUTION: distributes calls to different logical ends of the SIB based on user-specified parameters.
- LOG CALL INFORMATION: logs detailed information for each call into a file. The collected information may be used by management services (e.g. statistics) and not by call-related services.
- QUEUE: provides sequencing of IN calls to be completed to a called party.
- SCREEN: performs a comparison of data attributes against a filtered list of data attributes to determine whether the proposed values have been found in the list.
- SERVICE DATA MANAGEMENT: enables action on service data (i.e. to be replaced, retrieved, incremented, decremented, stored and deleted).

  This SIB has enhanced its capability to handle a pre-defined series of data manipulation actions, known as entry method, a new capability introduced in IN CS-2.
- SERVICE FILTER: filters the number of calls related to IN-provided service features. Such filtering will be based on user-specified parameters, such as Service Key, Destination Number. Filter Response can be reported to the service logic. The name of this SIB was "Limit" in IN CS-1 Recommendations but re-named "Service Filter".
- TRANSLATE: determines output information from input information.
- USER INTERACTION: allows information to be exchanged between the network and a call party, where a call party can be either a calling or a called party.

  This SIB is enhanced for the user interaction using out-band channel information (e.g. FACILITY IE) and for handling the User Interaction script which is a sequence of user interaction procedure. The Stage 1 definition of this SIB (SSD, CID, etc.) is enhanced to be able to handle the Component transferred between a user and the IN CS-2 network.
- VERIFY: provides confirmation that information received is syntactically consistent with the expected form of such information.
- BASIC CALL PROCESS: a specialized SIB which allows access to IN service/service features represented through the use of chains of SIBs. The interface points between this SIB and GSL are described as Points of Initiation (POIs), Points of Return (PORs) and Points of Synchronization (POSs) (refer to 4.5.2.1 and 4.5.6/Q.1223).

  BCP definition has enhanced its interface points to reflect the IN CS-2 requirements of interaction between the basic call process and service logics to allow more interaction points.

Annex A/Q.1223 provides a table showing relationships between SIBs and SIB operations.

7.2 Detailed description

This subclause describes useful information for IN CS-2 users such as general guidelines to use some specifications, example cases where some specifications are applied, or detailed protocol aspects, and so on. Descriptions in this subclause will supplement other IN CS-2 Recommendations.

7.2.1 Service capabilities

7.2.1.1 Examples of service with multiple instances of Single Point of Control

In IN CS-1/CS-2 compliant networks, the combination of services in one SSF is possible as described in the following examples.

The first example is valid for IN CS-1 and CS-2, whereas the second example is valid within the scope of IN CS-2 and beyond.

1) Charge Card service to a Premium Rate Number: The Charge Card service is invoked at the digit analyse information DP, and the Charge Card service logic carries out authentication, and collection of a B-party number. The call is then continued using the B-party number; since the B-party number is a Premium Rate number, a new control relationship is instantiated, and the analyse information trigger detects the Premium Rate service. The Premium Rate service logic then carries out the call distribution, number translation and special charging criteria. These two control relationships may be instituted in the same CCF/SSF.

2) A Charge Card call to a number with IN-based (CPH) Call Waiting: The Charge Card service is invoked at the digit analyse information DP. The Charge Card service logic then carries out authentication and collection of a B-party number. The call is then continued using the B-party number. As the B-party number is to a destination with IN-based Call Waiting, the Terminating BCSM will detect the Busy state of the destination at the T_Busy DP and the Call Waiting service will be invoked. These two control relationships may be instantiated in the same CCF/SSF.

7.2.2 Distributed functional plane

7.2.2.1 General consideration on internetworking between an IN structured public network and private networks

Both types of networks, public and private, offer telecommunication services to their users.

The need for interworking arises when the user communicates across boundaries between public IN structured networks and private networks.

If the internetworking between private networks and IN structured public networks becomes necessary, there are several alternatives. One is to implement similar IN architecture in the private network; another is to make available in public networks some functions that provide access interface to those IN functions. Of course other solutions may exist. The second approach is dealt with in 3.3 and 3.4/Q.1224. See these subclauses for information pertaining to interworking with non-IN structured networks.

Although the private networks involved may have different access types, e.g. ISDN or PSTN, and different levels of IN structuring (full, partial and no IN structuring) services must be provided to users in a consistent way. This involves cooperation of the networks to process and manage the services.

Private networks have similar functions as defined in public IN structured networks but different architectures are found in private networks, e.g. centralized, decentralized. In non-IN structured private networks, service logic and service data does not have to be separated from basic call processing.

In the case where the private network is not IN-structured, peer or equivalent functions are assumed in the private network.

7.2.2.2 SSF/CCF model

7.2.2.2.1 Multiple CDPN processing

In IN CS-2, the originating basic half-call state model, O-BCSM, is modified so that multiple "called party number (CDPN)" parameters sent from an SLPI are handled appropriately. Transition from the O_Alerting PIC to the Select_Route PIC is added and description for some PICs are modified for this purpose. If route_failure event is detected at Send_Call PIC or O_Alerting PIC, the BCSM transits to Select_Route PIC and some route_failure conditions on multiple CDPN parameters are checked at this PIC. If these conditions are met, the BCSM transits to Analyse_Information PIC and the next CDPN is processed (refer to 4.2.2.1.4, 4.2.2.1.5, 4.2.2.1.7 and 4.2.2.1.8/Q.1224).

7.2.2.2.2 Impacts of Reconnect operation on PICs

7.2.2.2.2.1 Operation sequence when the controlling party goes "on-hook"

Figure 7.1 shows an example of the case where "Reconnect" operation processing is required at the SSF/CCF. In this example, party "c" goes "on-hook", and SSF sends a report of ODisconnect or TSuspended (as appropriate).

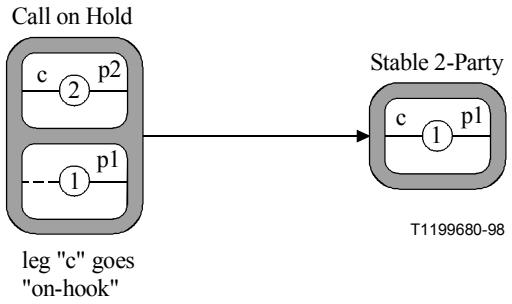


Figure 7-1/Q.1229 – An example case where "Reconnect" operation is needed

In this case, the SSF responds with the following message sequence to effect the transition from the "Call on Hold" CVS (Connection View State) to the "Stable 2-Party" CVS, after the controlling party goes "on-hook".

DisconnectLeg (p2) + MergeCallSegments + Reconnect

NOTE 1 – There is an alternative message sequence shown in the following:

ReleaseCall (CSID "2") + Reconnect

However, this approach has the following disadvantages:

a) It is not clearly understood how the call on hold (CS "1", with a "shared" controlling leg) will interact with the request to release the associated call (CS "2").

b) Which of the resources associated with the controlling leg would SSF/CCF clear when it processes the ReleaseCall (CSID "2") operation? The answer to this question may be implementation-dependent.

NOTE 2 – These operation sequences would also apply to the other two transition examples illustrated in 4.3/Q.1224.

The following subclauses include consideration on:

– impacts of "Reconnect" operation on BCSM; and

– possible Reconnect support for ISDN users.

7.2.2.2.2.2 Call Processing Model

IN CS-2 specification defines no new PICs or DPs for the processing of the Reconnect operation. However, a new sub-state ("Re_Ring") to the O/T_Active and O_Suspended PICs are considered for this processing. Figures 7-2 and 7-3 capture the modelling of the internal "Re-Ring" sub-state.

The states internal to the existing PICs are illustrated with dashed lines. When SSF/CCF receives the Reconnect operation to request reconnection of the controlling leg with the held party, the call on hold may be in the O_Active, O_Suspended or T_Suspended PIC. Call processing then moves into the internal sub-state entitled "Re_Ring", alerts the controlling leg (via power ringing and/or display information), sets the reconnect timer, and awaits a reconnect indication from the controlling leg. Expiration of the reconnect timer is considered an exception event, and the call is cleared. However, if the controlling leg reconnects within the allotted time, then call processing moves to the

O/T_Mid_Call DP (via the indicated PIC). That is, the "Reconnect Success" event may be visible to IN as a mid-call event.

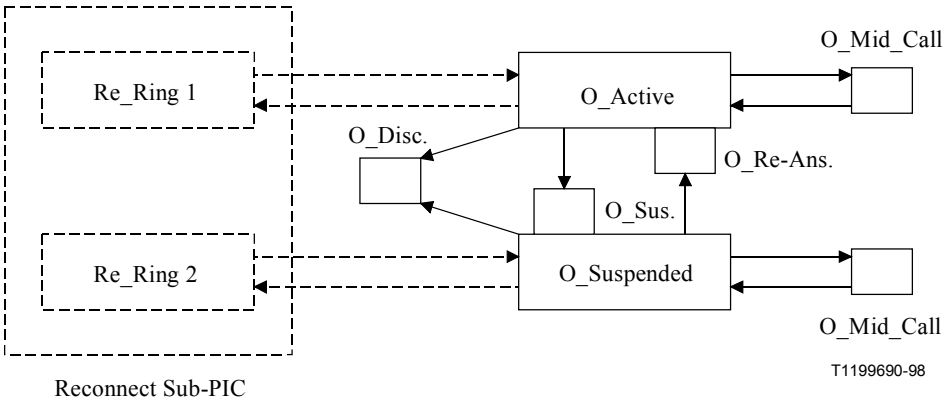


Figure 7-2/Q.1229 – "Re-Ring" sub-state in the O-BCSM

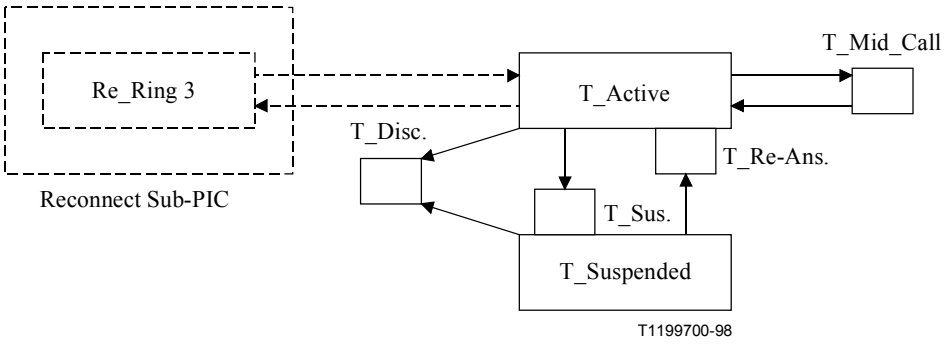


Figure 7-3/Q.1229 – "Re-Ring" sub-state in the T-BCSM

7.2.2.2.2.3 ISDN considerations

The Reconnect operation should also provide support for an ISDN controlling leg. A recommended approach is the addition of the "DisplayInformation" parameter to the "Reconnect" operation, in order to provide better support for an ISDN user.

7.2.2.2.3 BCSM transitions related to Call Waiting

The following provisions and transitions may be used to support the offering of "Call Waiting" to an originating caller where the originating call is in various set-up phases.

- If the originating call is between O_Null PIC and Originating_Attempt_Authorised DP, the Originating call should transit to O_Abandon DP, the originating call attempt is abandoned and the Waiting Call at the T_BCSM will proceed from T_Busy DP to Present_Call PIC and T_Answer DP in the expected way.
- If the originating call is between Collect_Information PIC and the (new) O_Term_Seized DP the Originating call continues uninterrupted into the O_Alerting PIC (Stable State). During the transient call progressing phase the Waiting Call should be queued. An Announcement could be played to the Calling party on the Waiting Call.

- If the originating call is in the O_Alerting PIC (Stable State), or when it reaches it (see point above) the option should exist for the Originating call to be interrupted with the Call Offer indication from the Waiting Call (during the initial call's Alerting Phase). If the Originating user who receives the Offered Call invokes the Call Waiting, the Alerting call will be put on Hold in the O_BCSM. To support this requirement, is was agreed that the O_Alerting PIC requires an exit and re-entry condition to O_Mid_Call.

  The Waiting Call at the T_BCSM will proceed from T_Busy DP to Present_Call PIC and T_Answer DP in the expected way.

  If the Held B_Party answers the initial call during the On Hold condition, the T_BSCM will proceed from T_Alerting (On Hold) to T_Active (On Hold) via the T_Answer DP, these events being reported to the O_BCSM in the usual manner. The Held B_Party must have a comfort Announcement played as soon as the T_Answer DP is encountered from T_Alerting (On Hold) PIC.

  The user may toggle between the two calls in the usual way.

7.2.2.2.4 Multiple retriggering

The IN CS-2 Recommendations do not specify a method to limit the number of times a call can retrigger. It also does not specify any method to prevent retriggering where the information received in a response contains the same as the criteria for the original trigger. For example, if the trigger criteria at the Analysed_Information DP is 555-1111, and the SCF returns in the Destination Routing Address a CDPN of 555-1111, the call will trigger at the Analysed_Information DP again. This effectively produces an infinite loop.

The following text provides two possible solutions to the problem (others may be possible), but none of these are mandatory.

1) *Letting the SCF prevent multiple retrigger*

   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 criteria was the Called Party Number. If the trigger criteria was something else, such as Shared Interoffice Trunk, then returning the same Called Party Number is not an issue.

2) *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).

7.2.2.3 SRF model

7.2.2.3.1 SRF Enhanced functions

This subclause presents all the possible actions the SRF is able to perform. We try to give them a standard input and output. We begin by describing the elementary actions which are combined together to build the enhanced functions.

7.2.2.3.1.1 Elementary actions

We found three elementary actions:

– Prompt playing: The purpose is to play a message, followed by a silence of adjustable duration. This can be interrupted by DTMF detection, speech detection, hang up, or an external message from a distant unity.

– Speech recording: The purpose is to record in a file the user's voice during a specific time and as long as no event occurs: (DTMF detection, silence detection, hang up, external message) or the end of allowed recording time.

– Data manipulation: such as operation on numbers, lists, strings or tables. The purpose is to control the number of prompts repetition or the data format.

7.2.2.3.1.2 Enhanced functions

The purpose is to define some standard enhanced functions, but not the entire set of enhanced functions. The enhanced function has to be customized to the service. They should be adapted to the current context of the call (human factors choices, ...) independently of the global service logic (the SCF).

7.2.2.3.1.2.1 Information diffusion

This enhanced function indicates to the user the state he is currently in. The information can be played several times with a variable degree of details.

The possible exits are:

– dtmf action (Result = name of DTMF);

– speech detection (Result = Detection);

– end of work (Result = End);

– external event (Result = External).

7.2.2.3.1.2.2 Get an information

This enhanced function allows the user to enter an information like a card number, a pin code or a phone number to be called. This allows procedure cancellation and error recovery.

It is divided into three phases:

– Prompts user to dial a number. In this case the user can cancel by pressing "star". The enhanced function exits with Result = Cancel.

– The Dial Number phase. The SRF is waiting for the keyed-in DTMF. Depending on the input, the enhanced function can:

  • exit with Result = OK and the Number is given to the SCF;

  • go to the Error Handling phase, if some errors occur;

  • return to the Dial Number phase.

The Error Management phase monitors the number of errors. If the user exceeds the number of attempts allowed, the SCF closes the user interaction. If not, the user may be informed of his mistake and be invited to prompt another number.

7.2.2.3.1.2.3 Speaker Verification (SV)

Speaker Verification is the process of verifying a person's claimed identity by analysing a sample of that person's speech. This form of security is based on the premise that humans can, to some degree of confidence, be identified by their speech. For telephone-based applications requiring access authorization, speaker verification can be used to identify or validate the caller. Before gaining access, however, the caller is required to have previously enrolled in a reference database. This enrollment is typically accomplished by repeating a multi-digit password several times.

Many varieties of applications can benefit from Speaker Verification technology. For example, banks and other financial service companies can greatly enhance the security of existing telephone-based account access system. The technology provides secure access to all callers, including rotary telephone users, and performs verification transparently during ordinary transactions, rather than requiring entry of supplemental phrases or PINs. The secure access is obtained much more quickly and easily than is possible with commonly used or suggested methods requiring supplemental keypad entry of PINs or spoken entry of additional phrases for speaker verification.

An example of speaker verification decision strategy is shown in Figure 7-4.

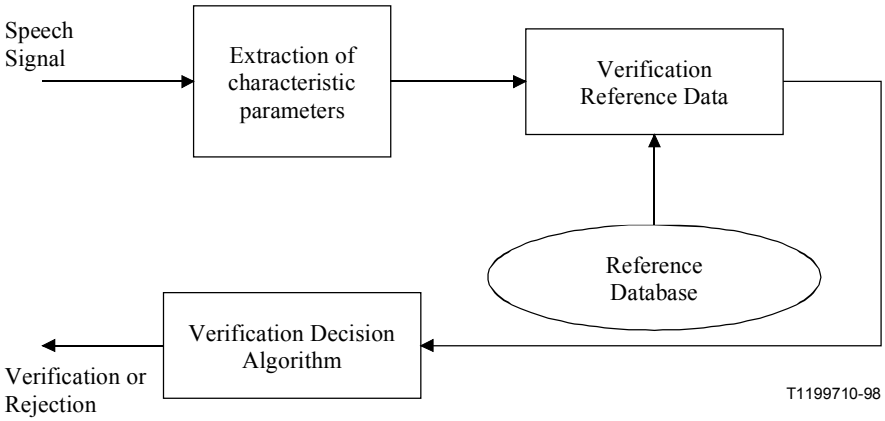


Figure 7-4/Q.1229 – Speaker Verification Decision Strategy

All reference parameters for each enrolled user are initialized on the first enrollment call, during which the password is typically spoken several times. Reference parameters could be gradually and cautiously updated upon each subsequent call. The reference parameters represent a sort of signature or "voiceprint" of each enrolled user; several other parameters are used to adjust the general strictness of verification. Some parameters are global in effect; others are user-specific, affecting only individual enrolled users.

An example of the implementation of this function could be a feature like Voice Identification. It allows users to place an outgoing call from anywhere, using their calling card. The user dials an access code and after the procedure of identification, by a keyword or general word, the user will be able to speak the number desired, or the name with procedures like VAD. Identification procedures are based on the Speaker Verification function in the SRF.

7.2.2.4 SDF model

7.2.2.4.1 SDF security

The X.500-series Recommendations (1993) provide some mechanisms to ensure the security of the communication between a DUA and a DSA. In IN, one of the communications to be protected is between the IN user and the SDF. Only the "bind" operation permits to embed mechanisms to authenticate an IN user to its provider database.

For the other security features between the customer and the database, the "execute" operation permits to offer them.

7.2.2.4.1.1 Security facilities examples

In several facilities, it is common to use mechanisms based on messages structured in two parts as follows:

– The first part is filled with data.

– The second part is the transformation of the first part by a symmetric (resp. asymmetric) cryptographic algorithm with a secret key (resp. private key).

We will use the following definitions:

verifier:

An entity which gets the assurance of an assertion by directly exchanging information with the prover or by getting an OK or a token from another authority.

Generally when two-part messages of the type described above are used, the verifier runs an algorithm to verify the conformance of the first part with public non-cryptographic properties (e.g. timestamp range, counter value, matching area, size, ...) and checks that the result of the transformation of the first part by a cryptographic algorithm has been done by an entity which knows a secret key (symmetric) or a private key (asymmetric).

prover:

An entity which makes an assertion to prove something (identity, authenticity of a message). Generally, the prover can make a two-part message (e.g. one-way authentication, certificated messages) or, if given the first part, the prover can provide the second part (e.g. two-way authentication).

authority:

An entity which provides the assurance of validity to a verifier. This authority is able to ensure a verifier of the validity of the prover but it might be unable to get this assurance for itself (e.g. untrusted verifier).

one-way authentication:

The data of the first part include a non-repeating number which is used to detect replay attacks and to prevent forgery. To avoid to have to store all the used numbers, it is common to use a counter or a number-including-time information. The second choice also permits to limit the lifetime of the authentication message, when many authentications could occur in the same laps of time (e.g. large time windows to take into account imprecise clocks or time transit) a random number is concatenated.

The verifier authenticates the prover after the following two verifications:

– The number has not be used before (e.g. check of counter or time range);

– In the case of a symmetric algorithm, the computation of the first part with the secret key as parameter matched the second part. In the case of an asymmetric algorithm, the computation of the second part with the public key as parameter matched the first part.

To reduce "Trojan horse" attacks, some contextual data, could be added to the first part of the message. Such as geographic information, transmission channel number. With these data a bad guy who induces a legitimate user to provide a valid message (e.g. by masquerading as a network) cannot use the stolen message elsewhere.

Similar mechanisms could be used to offer features such verification of a certificated message.

two-way authentication:

Several scenarios could be selected depending on the number of involved entities (commonly 2 or 3).

Scenario 1

The verifier is an entity of a service provider, the prover is the terminal of the user.

The verifier sends a random number, the prover answers, the verifier checks the answer.

When the protocol between the verifier and the prover is connectionless.

The verifier stores the sent random number, the answer message from the prover is a two-part message structured as the previously (data, result of a cryptographic transformation of these data).

The data will be the random received by the user, the second part of the computation of this random number by the user key contained by the terminal.

The verifier authenticates the prover after the two following verifications:

– the data is legal to the stored random number;

– in the case of a symmetric algorithm, the computation of the first part with the secret key as parameter matched the second part. In the case of an asymmetric algorithm, the computation of the second part with the public key as parameter matched the first part.

Scenario 2

The authority H which shares a secret key with the user (or knows the public key of the user) is an entity of the provider (e.g. a SDF).

A verifier V which trusts the previous verifier is another entity of the provider (e.g. an SCF), the prover is the terminal of the user.

The verifier V sends a random number to the prover, collects the answer of the prover and sends the random number and the response to the authority H; H returns an OK to V.

The authority H needs only to run the second verification:

– In the case of a symmetric algorithm, the computation of the first part with the secret key as parameter matched the second part. In the case of an asymmetric algorithm, the computation of the second part with the public key as parameter matched the first part.

In order that the authority H could be also a verifier, the random number has to be replaced as in the one-way authentication by a non-repeating number.

The data would be the non-repeating number and H will run, before the previous verification, the following verification:

– the number has not been be used before (e.g. check of counter or time range).

Scenario 3

The verifier asks a third party for information to carry on the authentication. This information could be the certificated public key or a precomputed couple (challenge, response).

The third party needs:

– an entry method to fill the first part (e.g. read the public key attribute, generate a random number, set up a session key); and

– a cryptographic algorithm to compute the result.

7.2.3 Intelligent Network Application Protocol (INAP)

7.2.3.1 INAP FSM interactions

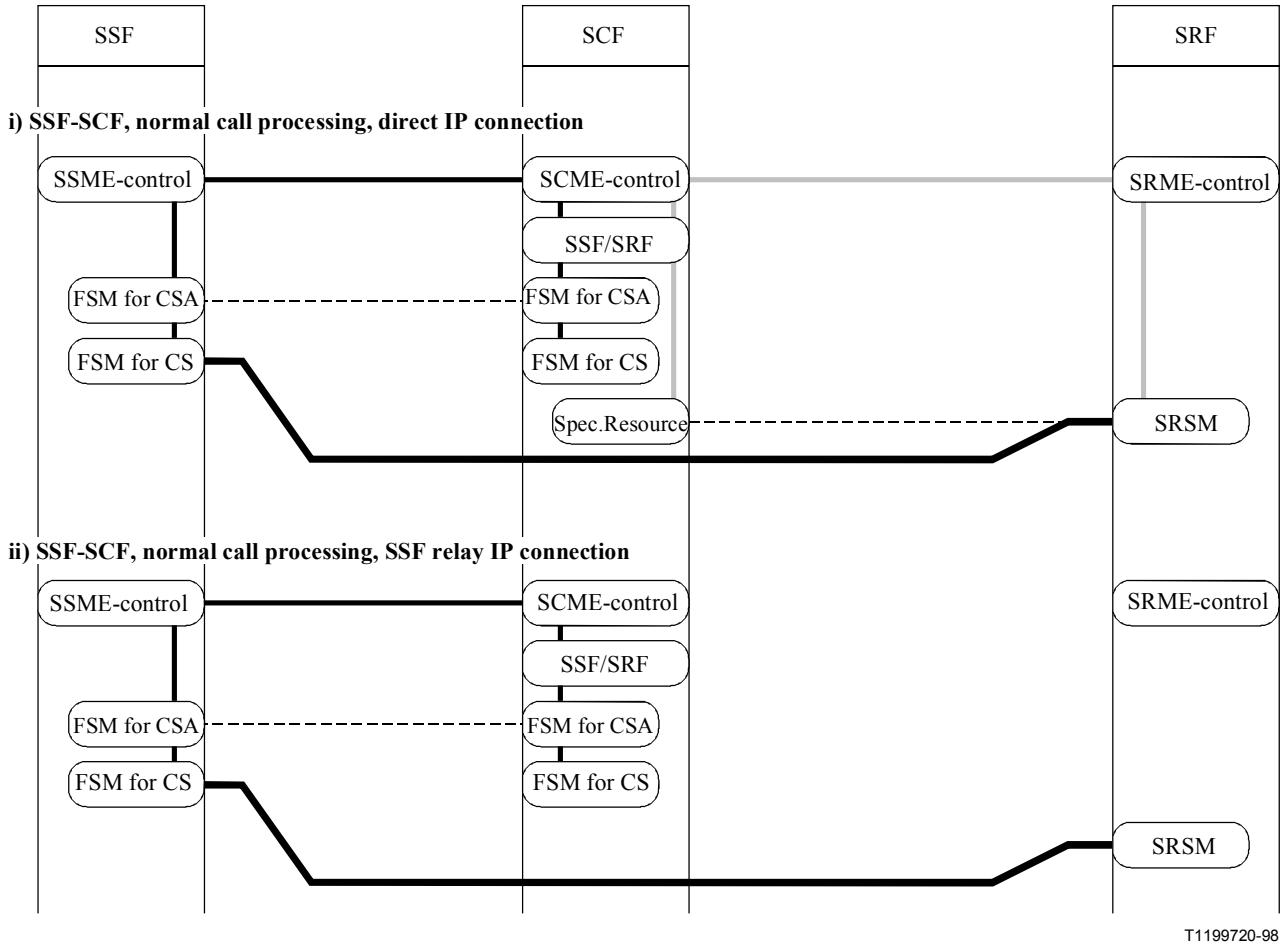
Clauses 11 to 15/Q.1228 define a number of FSM specifications to handle FE to FE interactions during call processing. This subclause contains diagrams which identify the relationships between the defined FEs during specific call processing conditions. In particular the diagrams identify:

a) the message path which operations pass through as they travel between FEs. This may involve a signalling stack (thin solid line) or a bearer connection (thick solid line).

b) the FSM level at which unique TC Dialogues (dashed lines) are established between a pair of FEs which use a signalling path to exchange operations.

7.2.3.1.1 SSF/SCF/SRF

Figure 7-5 shows the FSM interactions for SSF/SRF/SCF communication.



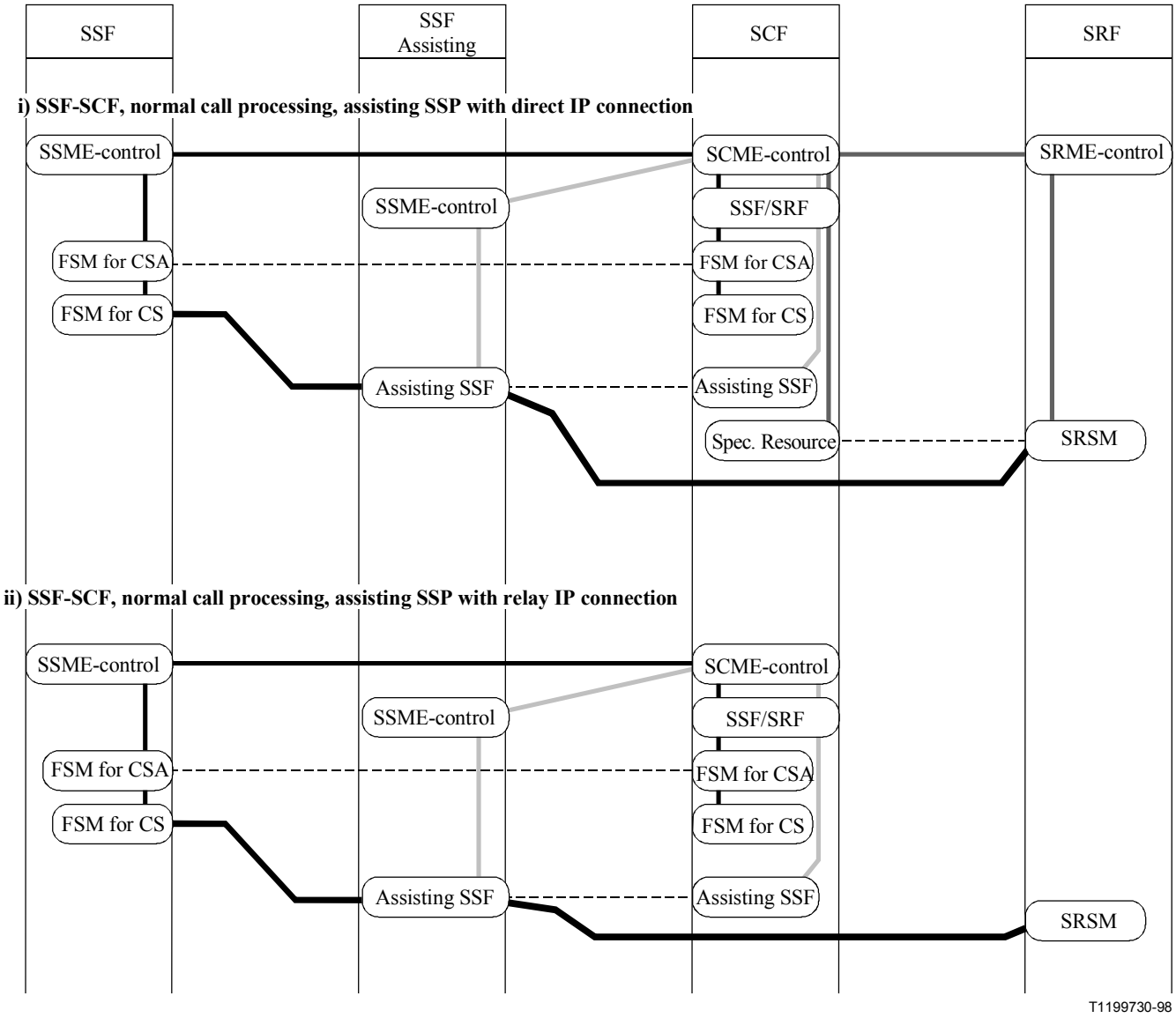
CSA Call Segment Association
FSM Finite State Machine
SCF Service Control Function
SRF Specialized Resource Function
SSF Service Switching Function

Message path
TCAP dialogue
Bearer connection

Figure 7-5/Q.1229 – FSM interactions for SSF/SRF/SCF

7.2.3.1.2 SSF/SCF/SRF with assisting SSP

Figure 7-6 shows the FSM interactions for SSF/SRF/SCF communication which utilizes an assisting SSF.



CSA Call Segment Association
FSM Finite State Machine
SCF Service Control Function
SRF Specialized Resource Function
SSF Service Switching Function

Message path

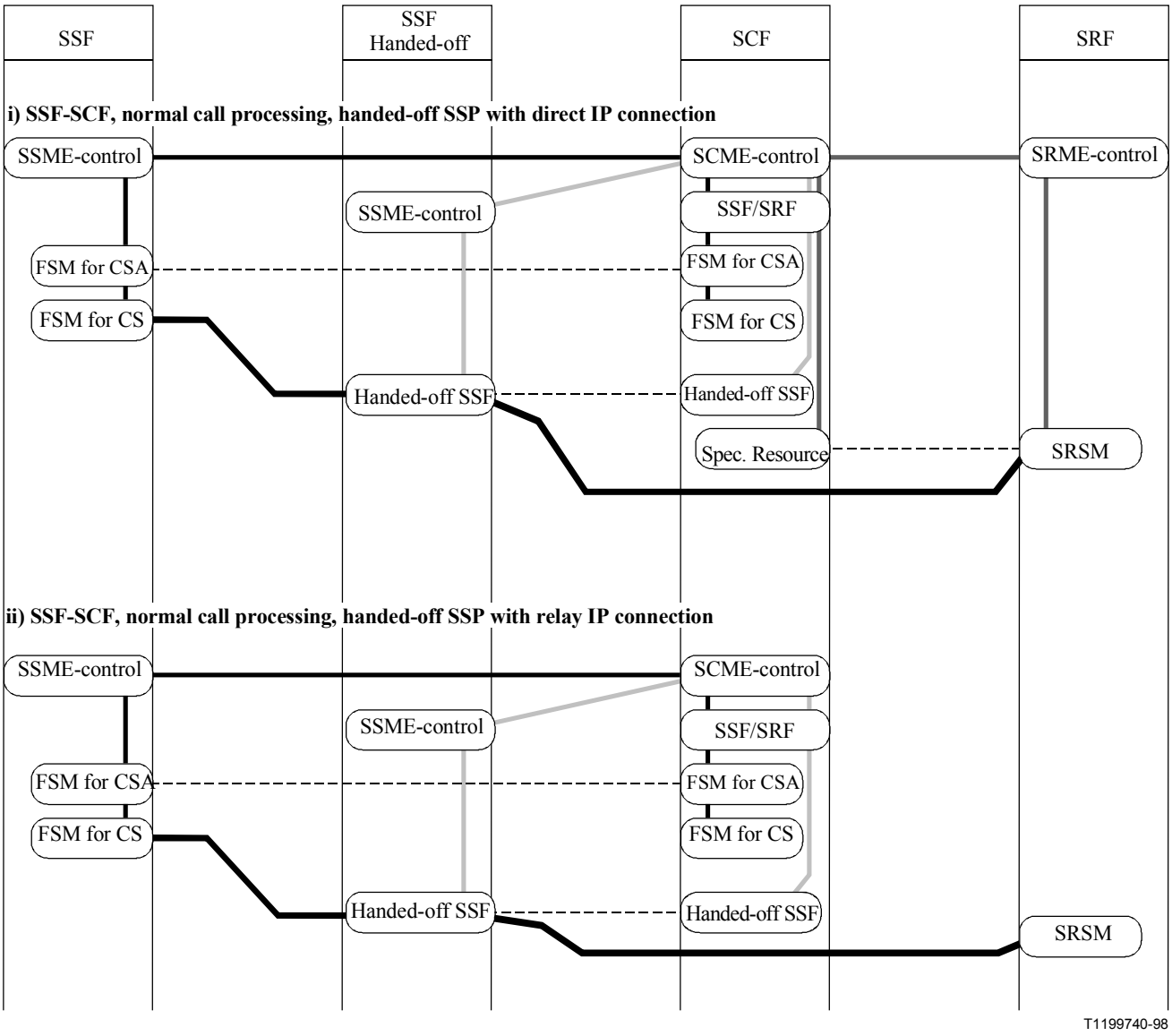
TCAP dialogue

Bearer connection

Figure 7-6/Q.1229 – FSM interactions for SSF/SRF/SCF with assisting SSF

7.2.3.1.3 SSF/SCF/SRF with handed-off SSP

Figure 7-7 shows the FSM interactions for SSF/SRF/SCF communication which utilizes a handed-off SSF.



CSA Call Segment Association
FSM Finite State Machine
SCF Service Control Function
SRF Specialized Resource Function
SSF Service Switching Function

Message path

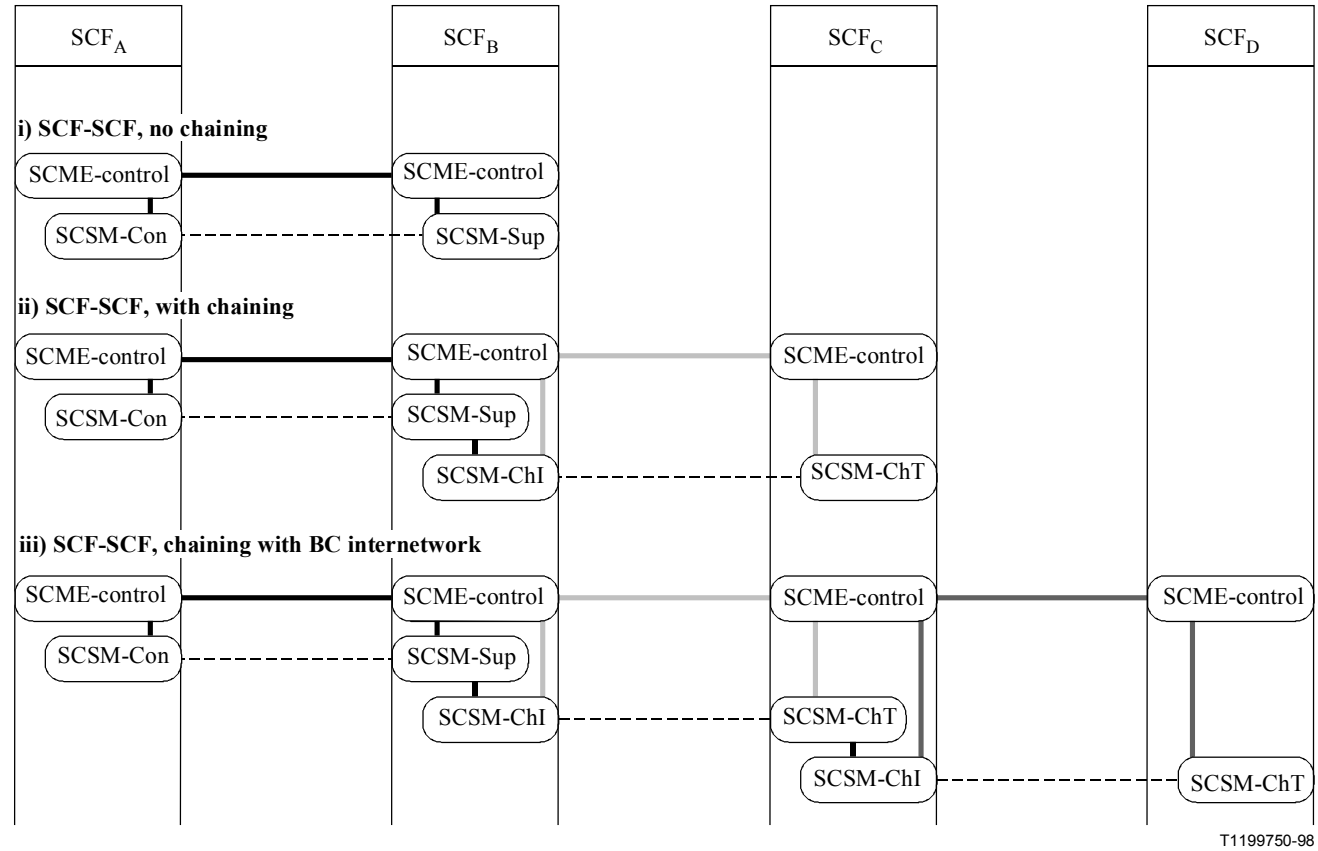
TCAP dialogue

Bearer connection

Figure 7-7/Q.1229 – FSM interactions for SSF/SRF/SCF with handed-off SSF

7.2.3.1.4 SCF-SCF

Figure 7-8 shows the FSM interactions for SCF/SCF communication.



CSA Call Segment Association
FSM Finite State Machine
SCF Service Control Function
SRF Specialized Resource Function
SSF Service Switching Function

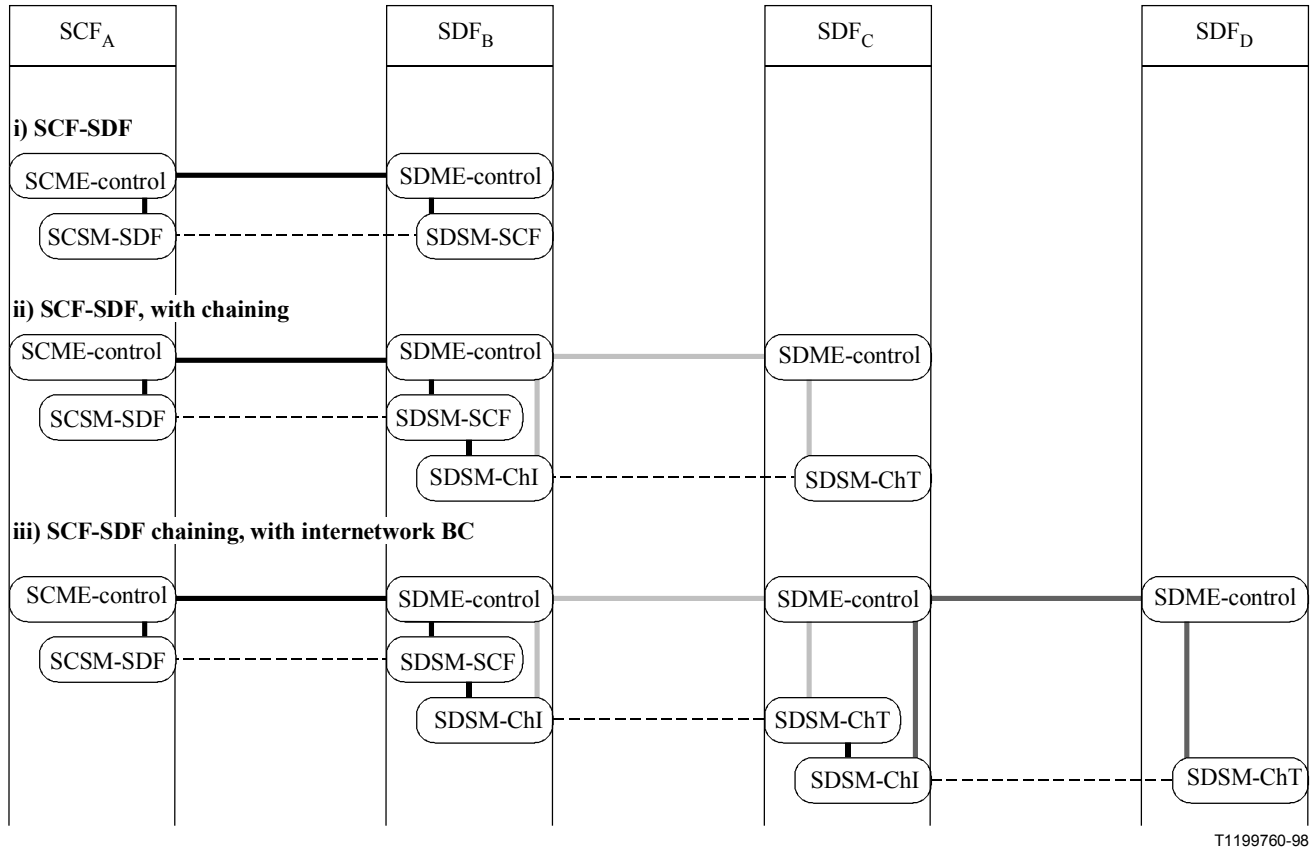
Message path

TCAP dialogue

Figure 7-8/Q.1229 – FSM interactions for SCF/SCF

7.2.3.1.5 SCF-SDF

Figure 7-9 shows the FSM interactions for SCF/SDF communication.



CSA Call Segment Association
FSM Finite State Machine
SCF Service Control Function
SRF Specialized Resource Function
SSF Service Switching Function

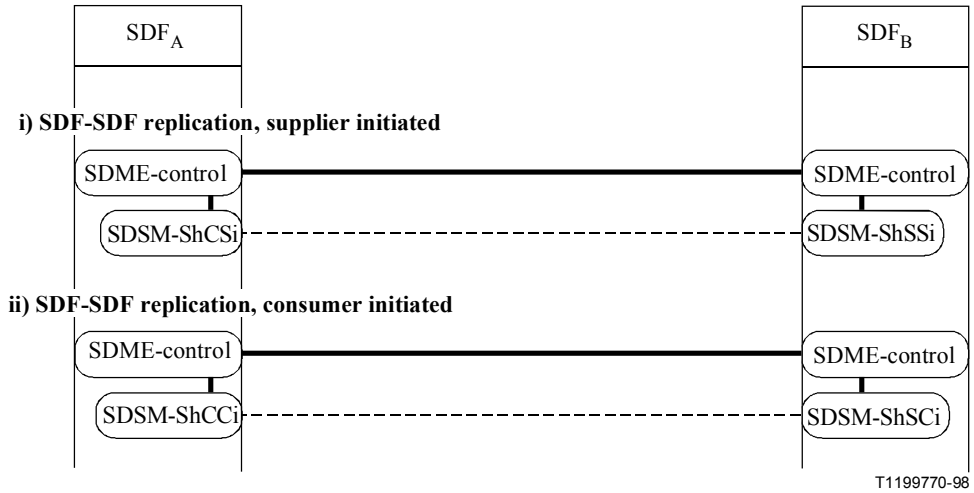
Message path

TCAP dialogue

Figure 7-9/Q.1229 – FSM interactions for SCF/SDF

7.2.3.1.6 SDF-SDF

Figure 7-10 shows the FSM interactions for SDF/SDF communication.



CSA Call Segment Association
FSM Finite State Machine
SCF Service Control Function
SRF Specialized Resource Function
SSF Service Switching Function

Message path

TCAP dialogue

Figure 7-10/Q.1229 – FSM interactions for SDF/SDF

7.2.3.1.7 Other

Figure 7-11 shows the FSM interactions for SSF/SCF management, SSF/SCF user-to-service and CUSF/SCF communication.

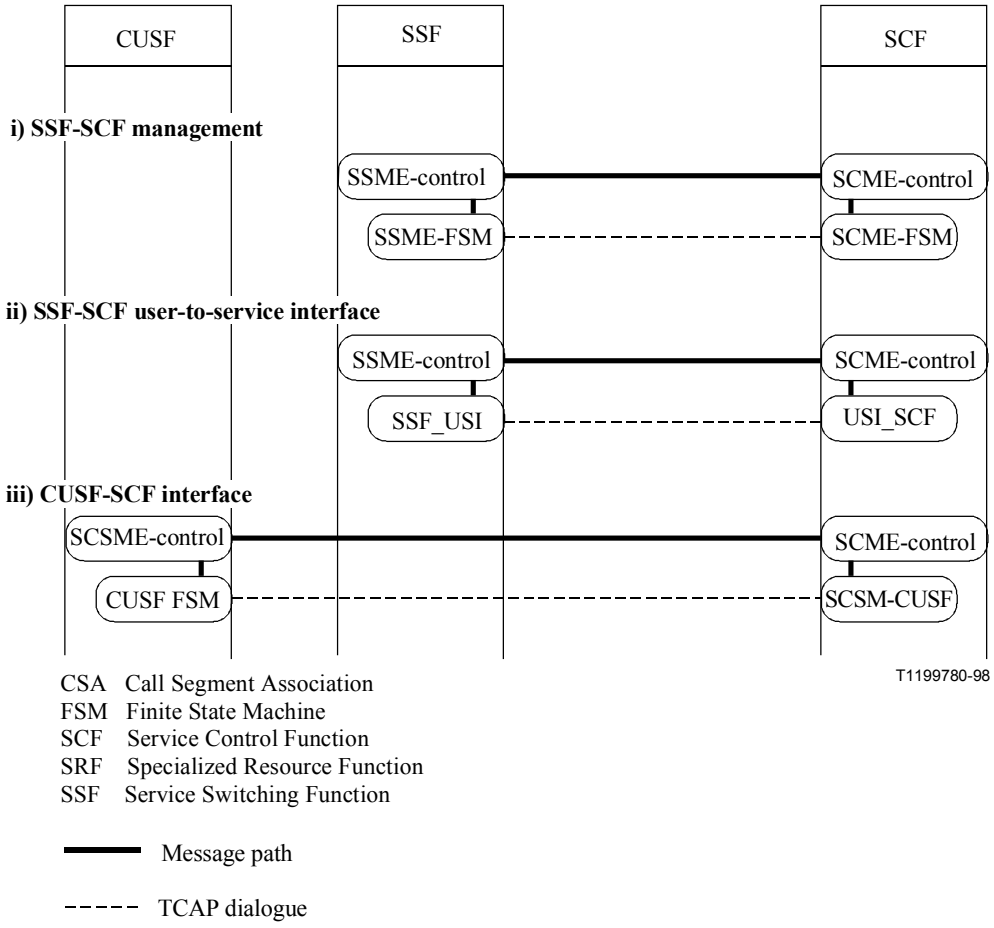


Figure 7-11/Q.1229 – FSM interactions for remaining FSMs

7.2.3.2 Example of use of the Out-channel Call Unrelated User Interaction feature

7.2.3.2.1 MSCs illustrating an example of use of the Out-channel Call Unrelated User Interaction

Several MSCs should be considered depending on the following points:

– The association is opened by the User or by the SCF.

– The association is released by the User or by the SCF.

1) *Opening of an association by the User*

The User initiates an association with the SCF as in Figure 7-12:

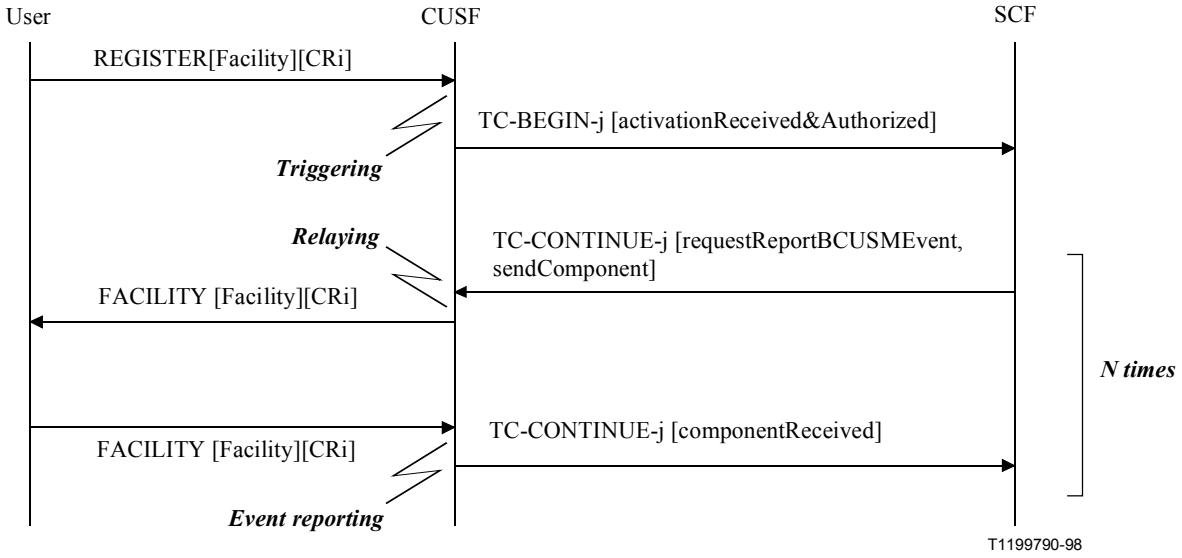


Figure 7-12/Q.1229 – Association establishment from the User

Later on, the User and the SCF dialogue using the Call Reference i between the User and the CUSF and the TCAP transaction j between the CUSF and the SCF.

2) *Opening of an association by the SCF*

The SCF initiates an association with the User as in Figure 7-13:

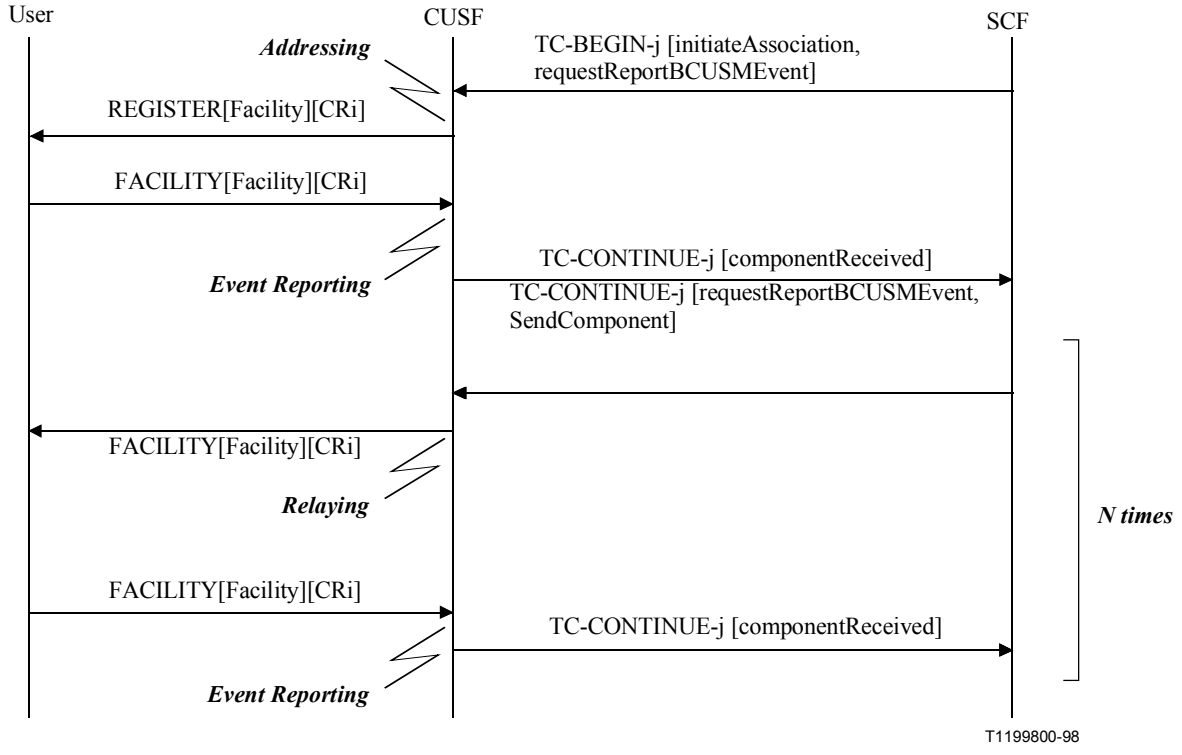


Figure 7-13/Q.1229 – Association establishment from the SCF

Later on, as in the first case, the User and the SCF dialogue using the Call Reference i between the User and the CUSF and the TCAP transaction j between the CUSF and the SCF.

3) *Release of an association by the User*

Two cases need to be considered depending on the signalling message sent by the User.

In the first case in Figure 7-14, the User releases the association (e.g. request the release of the association) sending RELEASE. On receipt of this signalling message, the CUSF sends associationReleaseRequested operation and waits for the releaseAssociation operation from the SCF.

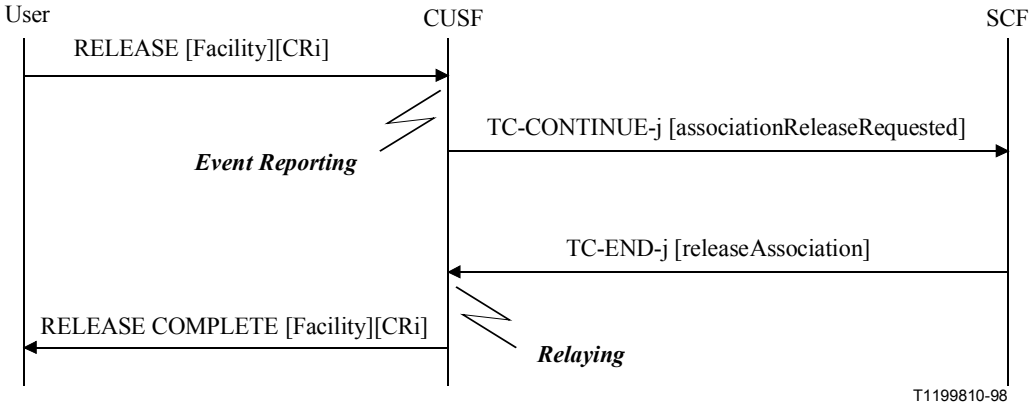


Figure 7-14/Q.1229 – Association release from the User (case 1)

In the second case in Figure 7-15, the User releases the association (e.g. request the release of the association) with the SCF sending RELEASE COMPLETE; the SCF can not do anything afterwards.

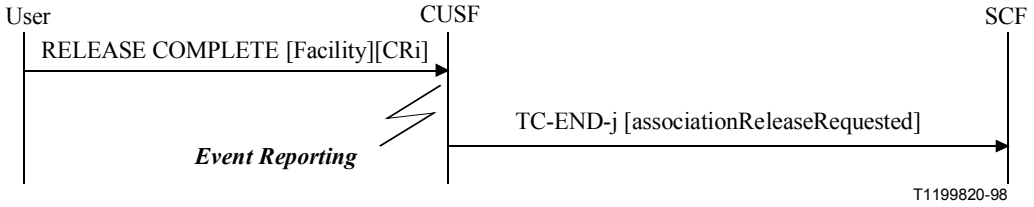


Figure 7-15/Q.1229 – Association release from the User (case 2)

4) *Release of an association by the SCF*

The SCF releases the association with the User as in Figure 7-16:

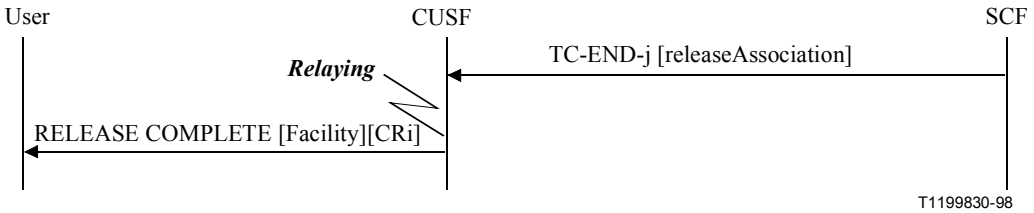


Figure 7-16/Q.1229 – Association release from the SCF

7.2.3.2.2 CUSF procedures

At the CUSF level, as indicated in the previous example, two different kinds of CUSF procedures should be defined: the first ones correspond to the initiation of a call unrelated "User-SCF" dialogue while the second ones correspond to the relay of information between the User and the SCF during an existing call unrelated "User-SCF" dialogue.

In the "User to SCF" direction, the CUSF procedure instantiating the call unrelated "User-SCF" dialogue is denoted the "triggering" procedure. In the reverse direction, it is denoted the "addressing" procedure:

– The addressing procedure consists in a correspondence between the Called Party Number and the targeted line identity.
– The triggering procedure consists in the analysis of the triggering criteria which may be a line-based criteria or a criteria embedded within the Facility IE received from the User.

In the "User to SCF" direction, the CUSF procedure relaying the information during an existing call unrelated "User-SCF" dialogue is denoted the "event reporting" procedure. In the reverse direction, it is denoted the "relaying" procedure. These procedures just ensure the relay of information between the User and the SCF. This relay might be fully transparent.

7.2.3.3 Security guidelines for Out-Channel Call Related User Interaction

7.2.3.3.1 Limitation of usage of SS7 signalling capabilities

In order to protect the ISDN network signalling system from being overloaded with user-to-service and service-to-user signalling:

– the maximum size of the User-to-service Interaction/Service-to-user Interaction (UTSI/STUI) data containers shall be limited;
– the rate of user-to-service signalling events shall be limited in the local exchanges; and
– the rate of service-to-user signalling events shall be limited in the SSPs.

In the case where the user terminal violates these limitations, the local exchange shall ignore the UTSI information element and in the case where the SCP violates these limitations, the SSP shall ignore the STUI information element.

7.2.3.3.2 Prevention of UTSI-feature usage by unauthorized users

The local exchange shall accept an UTSI information element in the following cases only:

1) *Call-related signalling*

The local exchange shall accept an UTSI information element in a call-related signalling message if:

– UTSI signalling is carried in a basic call control message;
– UTSI signalling is not carried in a basic call control message (e.g. FACILITY message) but has been explicitly allowed by an IN service logic for that particular call via INAP/ISUP signalling,

the latter capability may require additional ISUP signalling in forward and backward direction.

2) *Call-unrelated signalling*

The local exchanges shall accept an UTSI information element in a call-unrelated signalling message, if:

– UTSI signalling is allowed according to access subscription.

NOTE – In a particular network, the network operator may decide to allow call-unrelated UTSI signalling for every subscribed access. Such a decision may depend on charging decisions.

In all other cases, the local exchanges shall ignore the UTSI signalling.

The local exchange shall ignore any STUI signalling, if received from the user side.

7.2.3.4 Guideline for the use of advanced SCF-SDF searching and information modelling mechanism

7.2.3.4.1 Advanced SCF-SDF searching and information modelling mechanisms

The SCF-SDF interface is an agreed internetworking interface in IN CS-2, as in IN CS-1. The overall performance of IN service implementations will be heavily influenced by data performance and access performance. A number of factors will influence performance; however, a key determinant will be the number of protocol messages that are required over the SCF-SDF interface, particularly in the case of services which must support international roaming, where message delays can be very significant.

Performance will also be affected by the data structuring and whether there is a need to maintain data consistency. For example, where data is logically shared between different parts of a data hierarchy, there must be mechanisms to either locate the data in one place or to ensure that the data is set to the same value in each place.

To optimize the database with respect to these parameters requires intelligent use of the sophisticated searching and information modelling features of the SCF-SDF interface, as described in 7.2/Q.1228. These mechanisms include:

- aliases;
- multi-valued attributes;
- collective attributes;
- extensible matching rules;
- attribute contexts;
- entry methods.

1) *Aliases*

Aliases are entries that point to other entries in the database. An alias is used to provide alternative names for an object. This can be used to implement a consistent many-to-one mapping. The use of aliases has the following advantages and disadvantages:

- Aliases can be used to refer to entries which contain data common to many parts of the hierarchy, for example, number translation tables common to many service users. Data consistency is easily maintained.
- Information models which are not strictly hierarchical can be represented in a hierarchy using aliases.

2) *Multi-valued attributes*

Attributes of an entry, other than naming attributes, can have more than one value. This can be used to create a consistent many-to-one mapping. For example, consider an entry that represents a geographical region, and has an attribute which represents a valid calling line prefix for that area. If this region has more than one valid calling line prefix, then the calling line prefix attribute can have more than one value. A search on this attribute using any of the valid calling line prefixes will result in the correct data being returned. The use of multi-valued attributes has the following advantages and disadvantages:

- Less physical space is likely to be required to store a list of information using a multi-valued attribute than by using a set of subordinate entries. Sets of subordinate entries require naming, access control and other data to be stored for each element of the set.
- Multi-dimensional lists cannot be implemented with multi-valued attributes.
- Values within a multi-valued attribute cannot be shared between two instances of the attribute. Therefore data consistency may not be maintained.

3) *Collective attributes*

A collective attribute is an attribute that is common to all entries in a subtree. Collective attributes can be used to minimize the number of database accesses required to complete a complex search which must satisfy a number of search criteria at different levels in the hierarchy. For example, "locate all entries of type X with attribute A ≥ 1 and whose parent has attribute B ≤ 2". If the parent entry's attributes are made collective, and thus visible to the child entry, the search may be performed in one operation. Otherwise, searching the different levels of the hierarchy requires multiple searches. The use of collective attributes has the following advantages and disadvantages:

- Multiple searches down through a hierarchy can be compressed into one search.
- Subtree searches involving collective attributes may involve many, many entries and hence incur a significant performance penalty.

4) *Extensible matching rules*

Along with the built-in matching rules, the IN CS-2 SCF-SDF interface allows schema designers to add new matching rules to attribute types. Two applications of this mechanism are immediately obvious. The first is to search on components within structured attributes. For instance, an attribute may be of a user-defined type which contains both a user-name and password. A search may be required to match on an entry that has a given user-name only. A new matching rule can then be defined that compares only the user-name field.

Secondly, this mechanism allows an attribute to be defined in a number of different ways, such as time being defined as Greenwich mean time or local time.

The use of extensible matching rules has the following advantages and disadvantages:

- The schema designer has the flexibility to introduce new types and (re)define the matching rules accordingly.
- Matching rules only apply to search criteria and cannot be used for further customization.

5) *Attribute contexts*

An attribute context is information that can be attached to an attribute to define the validity of a attribute value. An attribute context is very similar to an attribute, having both a type and a value. Attributes which contain attribute contexts are by nature multi-valued.

A request, either directly or as part of a search criteria, to an attribute containing one or more contexts is handled in the following manner. If no context is provided, the SDF returns a value for the attribute based upon some hidden context specific algorithm. If a context is provided, in the form of a list of context values, then the SDF tries to find an attribute value that has the same value for its context.

The most useful example of an attribute context is that of a lifetime of an attribute. Others include a language context for announcement information. The use of attribute contexts has the following advantages and disadvantages:

- Attribute contexts can make search criteria much simpler.
- Attribute contexts allow the schema designer an enhanced level of customization, although still on an attribute basis.

6) *Entry methods*

Entry methods enable complex data manipulation at the SCF-SDF interface to be performed in a single database operation. Instead of many complicated database operations, a message is passed that contains all the input parameters to an entry in the hierarchy. The internal operations are then performed at the SDF including any logic required to link the operations together. The output values are then passed back to the SCF. The use of entry methods have the following advantages and disadvantages:

- The number of external database accesses is greatly reduced.
- The exact logic of the SDF operation is hidden, enabling service providers firstly to provide service differentiation whilst retaining a common interface, and secondly to provide smooth evolution of services.
- Interworking with non-IN and private network databases may become easier.
- Data privacy and security may be enhanced by appropriate hiding of the information model behind the interface method.

7.2.3.4.2 Service example

This subclause illustrates how advanced searching and information modelling mechanisms can be applied to a complex service. A number of scenarios are described as follows:

- The first is a simple non-optimized implementation.
- The second implementation minimizes the number of database accesses through use of advanced SCF-SDF searching and information modelling mechanisms, simplified algorithms and by making some assumptions on how the data is to be used. These assumptions may result in replicated data and the imposing of some limitations on how the data is described.
- The third implementation maximizes flexibility in defining service data, removes duplication of data, and removes the limitations of the second implementation. Despite the use of advanced searching and information modelling mechanisms, the number of database accesses remains large.
- The fourth implementation makes use of entry methods to minimize the number of database accesses, maximize the flexibility of data entry, and removes the limitations of the second implementation.

The four alternatives listed here are not the only options available, but provide an illustration of how and when to use the advanced searching and information modelling mechanisms.

7.2.3.4.2.1 The example service

The service we wish to implement involves the routing of a call based upon the calling line identity, time-of-day, day-of-week, percentage call distribution and multiple call destinations for busy outgoing lines. There are six steps to translate the number as follows:

1) Selection of a calling region using the A and B party numbers.
2) Selection of a day-of-week for the selected region.
3) Selection of a time-of-day for the selected day-of-week.
4) Selection of a destination end point, using a percentage call distribution (splay), for the selected time-of-day.
5) Retrieval of two outgoing numbers for the selected destination.
6) A call attempt to the first number. If this call fails, a second call attempt to the second number.

A simple decision tree for an instance of this service is illustrated in Figure 7-17.

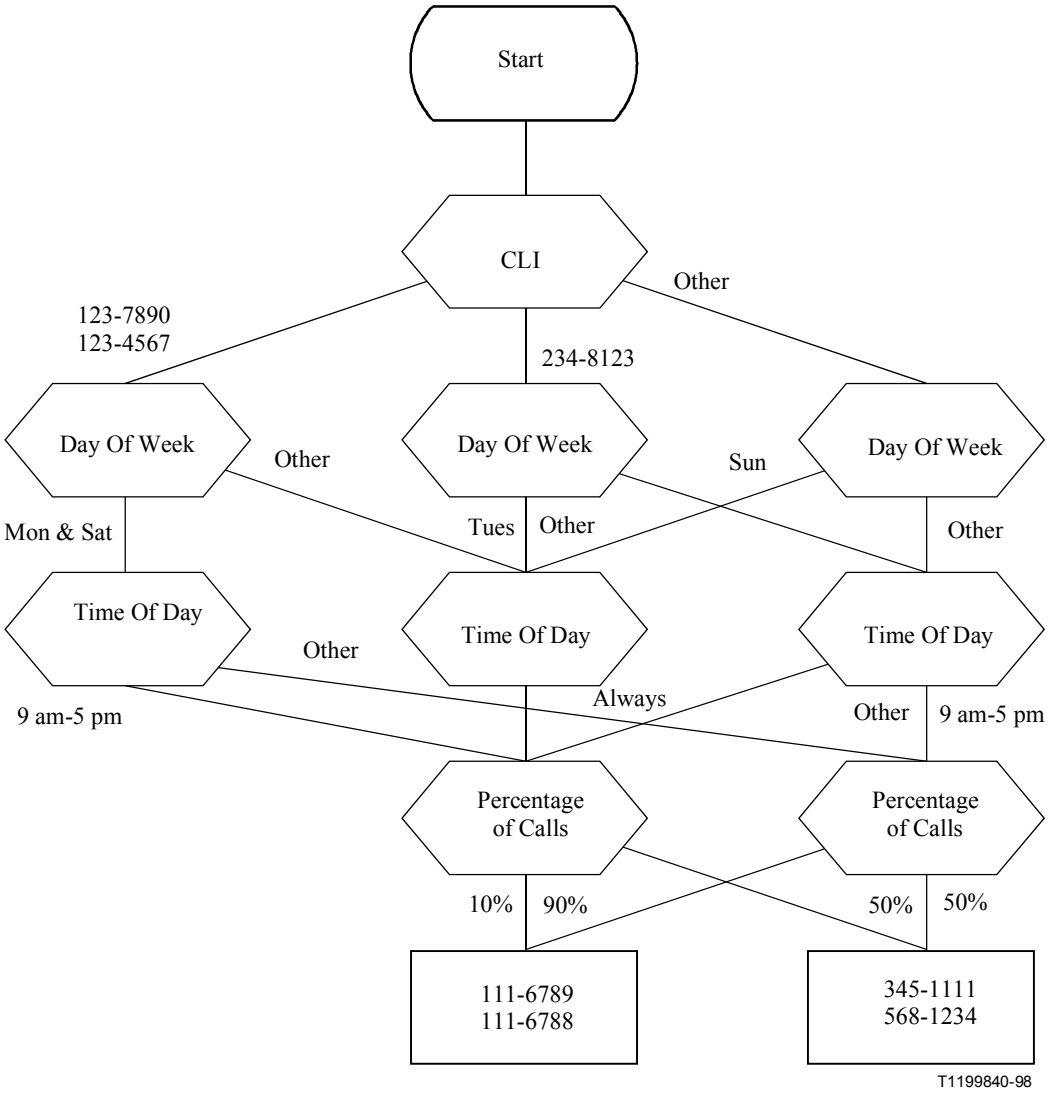


Figure 7-17/Q.1229 – Decision tree for service example

Depending on the implementation, some of these steps may be performed in a single operation, or in multiple operations.

7.2.3.4.2.2 A Simple implementation

In this solution we ignore the possibility of using advanced searching information modelling mechanisms, and use only the guidelines set out in Appendix II/Q.1218. The data schema will be discussed in the order in which it would be designed; the complete schema is illustrated in Figure 7-18. Likewise, the complete information flows are shown in Figure 7-18. In this implementation we will create an entry for each customer using this service. This object class is labelled "Customer". The naming attributes of Customer include a service identifier ("Service") and the dialled number ("BNumber").

The translation of a Calling Line Identity (CLI) to a region can be implemented by comparing a CLI to a list of entries of object class "MapE" that have both a "CLI" attribute and a "Region" attribute. MapE entries are placed below the Customer entry in the schema as shown in Figure 7-18. This would enable the SCF to interrogate the SDF using a standard one-level search (refer to the first information flow in Figure 7-19).

A set of day-of-week's for a given region may be represented by an entry of object class "Locality" placed below the Customer entry in the hierarchy. Below Locality, each day-of-week is represented by an entry of object class "DOWE", one for each day of the week. Locality and DOWE have naming attribute of type "Region" and "Day" respectively. A disadvantage of this hierarchy is that there can only be one day-of-week entry for each weekday, and thus there can be no exceptions for specific days. For instance, you may want a certain translation for all Thursdays, with the exception of Thursday, 25 December.

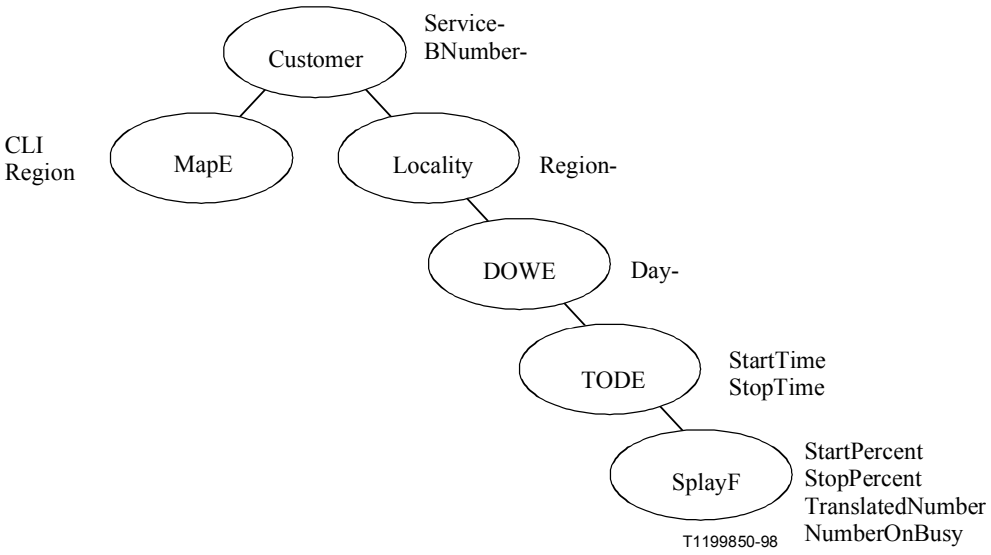
A set of times-of-day for a given day-of-week may be represented by an entry of object class "TODE" placed below a DOWE entry. The combination of fixed day naming (see above) and a hierarchy of Locality/DOWE/TODE will enable steps 2) and 3) above to be implemented in a single operation. Each TODE entry would have attributes of "StartTime" and "StopTime". A search can then be made to find an entry that is below the DOWE entry and has start and stop time values either side of the current time.

An entry of object class "SplayF" would need to be created to implement step 5). This entry would represent a set of rules based upon percentage distribution. A very crude algorithm could be used as follows:

- Each SplayF entry would have an attribute "StartPercent" and an attribute "StopPercent".
- The number of calls is kept in the SCF.
- The number of calls modulus 100 is compared to the percentage range for each entry.

This algorithm would be entirely unsuited to all but the largest customers.

If these assumptions are made, and the schema of Figure 7-18 is used, the information flow diagrams for the service would be as shown in Figure 7-19.



- = Naming Attribute

Figure 7-18/Q.1229 – Call routing service information model using simple searching mechanisms

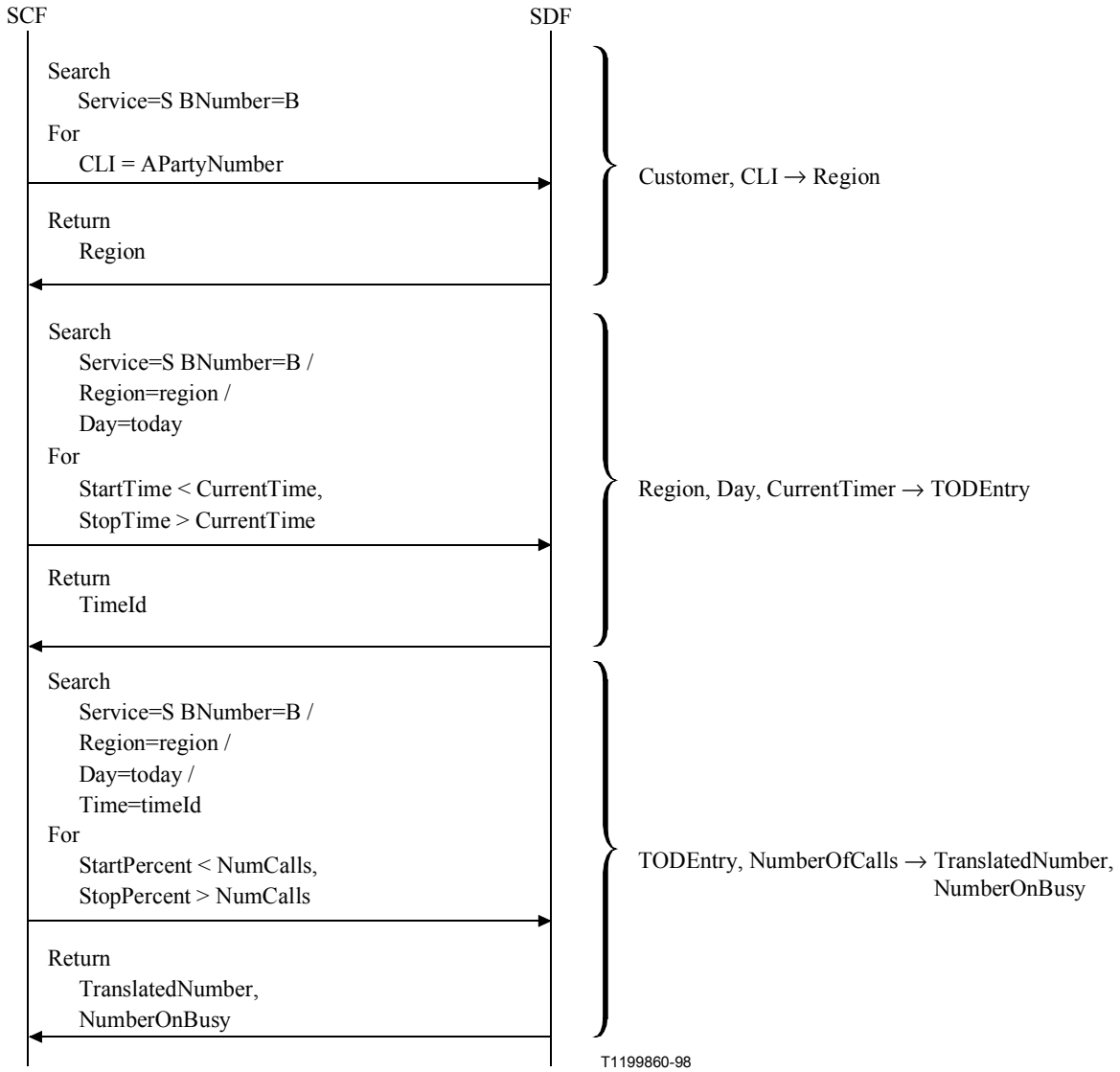


Figure 7-19/Q.1229 – Information flow diagram for service using simple searching mechanisms

Figure 7-19 assumes the distinguished name of the Customer entry is "Service = S BNumber = B", where B is the called number. Once the service has been triggered, an INAP message is sent from the SCF to the SDF containing a bind and this first search operation. This search operation translates the calling line identity (CLI) to a region. The region is then used with the service, B-number, day-of-week and current time to return a reference to a time-of-day entry. A subsequent search translates the number of calls to the translated number and alternate number. The total number of database requests for this service, ignoring binds and unbinds, is three.

7.2.3.4.2.3 Using advanced mechanisms to minimize database accesses

In this implementation we make use of the advanced searching and information modelling mechanisms to minimize the number of database access. The solution uses the same assumptions made in the previous implementation.

The first step of translating the CLI to a region is simplified through the use a multi-valued attribute. Instead of two classes, CLI and Locality, a single Locality object class is created with a multi-valued CLI attribute. This enables each Locality entry to be identified by more than CLI using a search operation. This has the added effect of maintaining referential integrity between the CLI and the locality.

The use of a multi-valued CLI list also enables all of the parameters required for the translation, namely the CLI, the time-of-day, day-of-week and percentage of calls, to exist in the superiors of the entry that contains the two translated numbers. If all of these parameters are made collective attributes, then they theoretically exist in the entry at the bottom of the hierarchy. This means a single search operation can be used to identify the final entry. Although this greatly reduces the number of database accesses, it would most likely increase the access time, since the search would be performed on all entries in the subtree below the "Customer" entry.

The data schema is shown in Figure 7-20 while the information flow diagram for this implementation is shown in Figure 7-21.

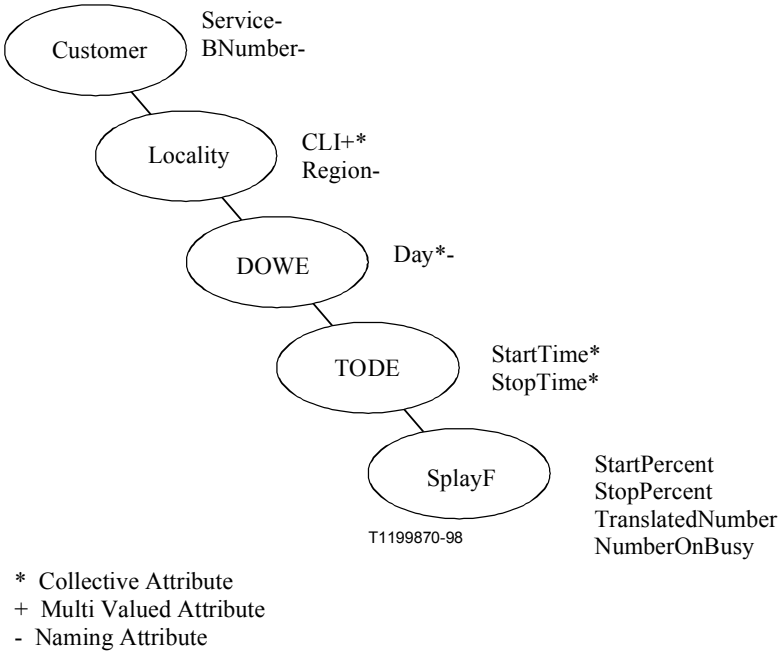


Figure 7-20/Q.1229 – Call routing service information model using multi-valued and collective attributes

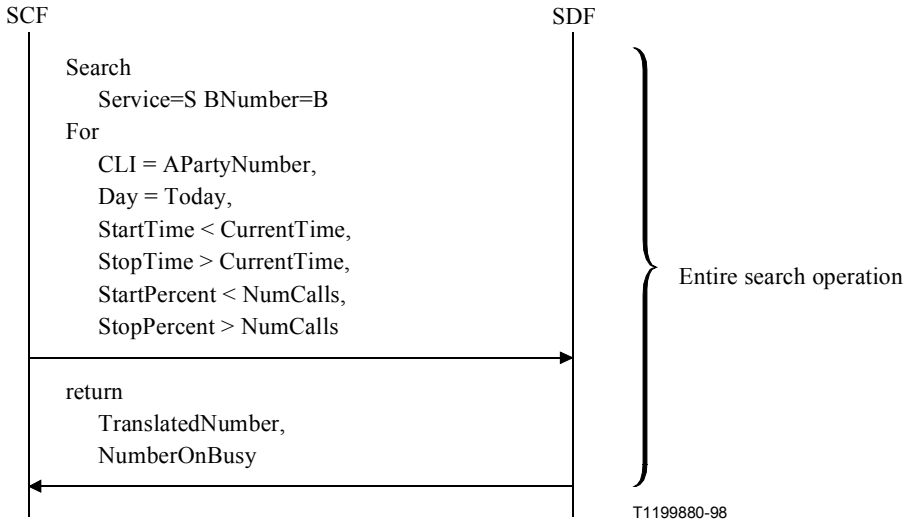


Figure 7-21/Q.1229 – Information flow diagram for service using multi-valued and collective attributes

The total number of database requests for this service, ignoring binds and unbinds, is one.

7.2.3.4.2.4 A solution to maximize flexibility

Customer requirements are such that they expect more flexibility than the assumptions imposed on them in the two previous implementations. The following improvements may need to be made to the service:

- The algorithm for distributing calls based on a percentage distribution needs to ensure that calls are distributed according to the specification even for very small numbers of calls.
- The data for day-of-week routing needs to be more flexible so that both general terms, such as every Monday, and specific terms, such as Monday the eighth of January 1996, can be defined.
- The data for time-of-day routing likewise needs to handle both general rules and exceptions to the rules.
- Due to the large amounts of memory and cost of defining complete CLI-to-region maps, maps should be shared between customers wherever possible.
- Day-of-week entries need to be able to be shared between different localities.
- Time-of-day entries need to be able to be shared between different day-of-week entries.
- Percentage routing parameters need to be able to be shared between different time-of-day entries.
- Numbers lists need to be shared between routing destinations.

Figure 7-22 illustrates the data model of a solution that satisfies these requirements.

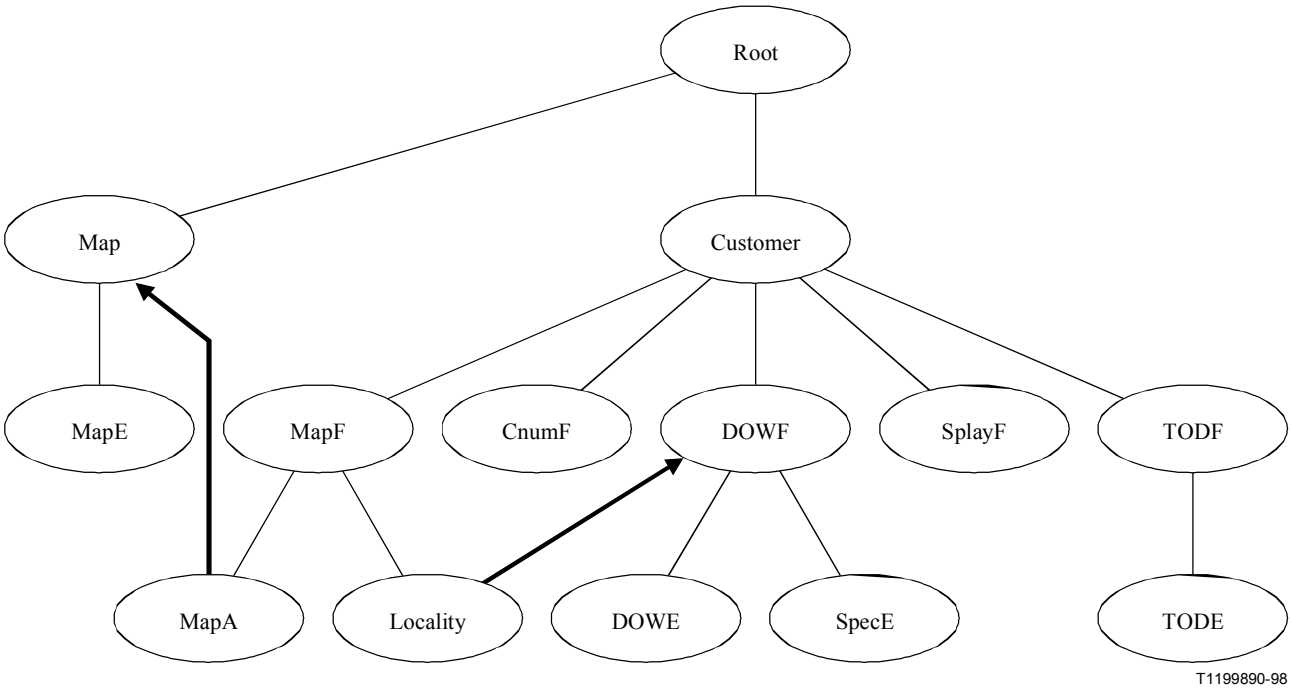


Figure 7-22/Q.1229 – Information model for maximum flexibility

In this example each set of rules, or feature, such as time-of-day routing can be shared as outputs from other features. This means that each feature has to be translated one at a time. The first five steps listed in 7.2.3.4.2.1 are performed as separate sequential operations with the output of each

being used in subsequent operations. These five steps are listed in the following subclauses. All use the schema defined in Figure 7-22. The schema diagrams used in the following subclauses are more detailed diagrams of portions of Figure 7-22 with additional attribute information.

Figure 7-23 shows in more detail the schema used in translating the A party number to a region. Figure 7-24 shows the information flows used to implement the operation.

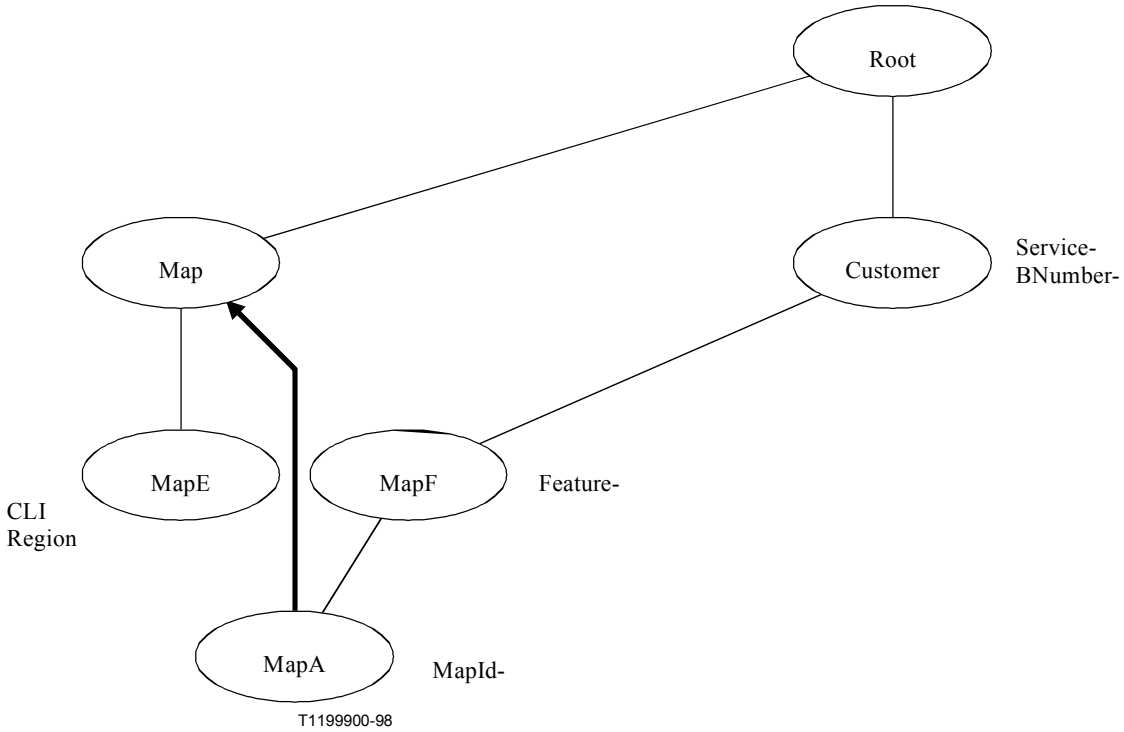


Figure 7-23/Q.1229 – Information model to translate the CLI to a region

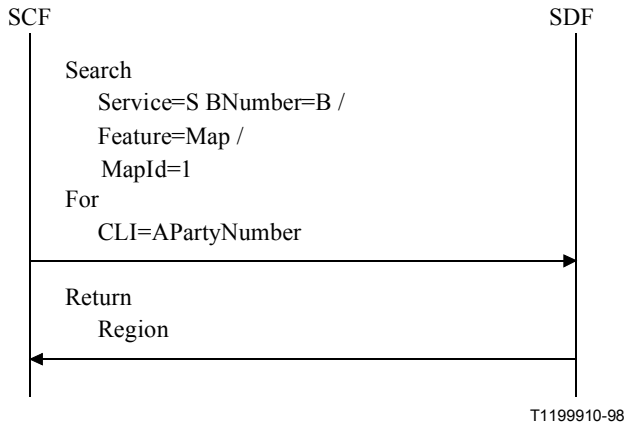


Figure 7-24/Q.1229 – Information flows used to translate the CLI to a region

The information flow diagram does not include the bind. In this example the Customer naming attribute is once again "Service = S BNumber = B". Each customer has an alias to a shared CLI map. The region associated with the Map entry is returned if the search is successful.

Once the region is retrieved from the maps, the day-of-week information is retrieved by using the appropriate region alias entry listed with the map feature: the current day needs to be translated to a time-of-day feature. The process of retrieving the region first, followed by the day-of-week information, is required because the map information is shared between many customers. By using an OR in the filter, the current day can be compared to a list of weekday entries and a list of special days in the one search. Note that the lists could also be stored as a multi-valued attributes with application contexts instead of a list of subordinate entries. Figure 7-25 below shows in more detail the schema used in this operation. Figure 7-26 shows the information flows used to implement the operation.

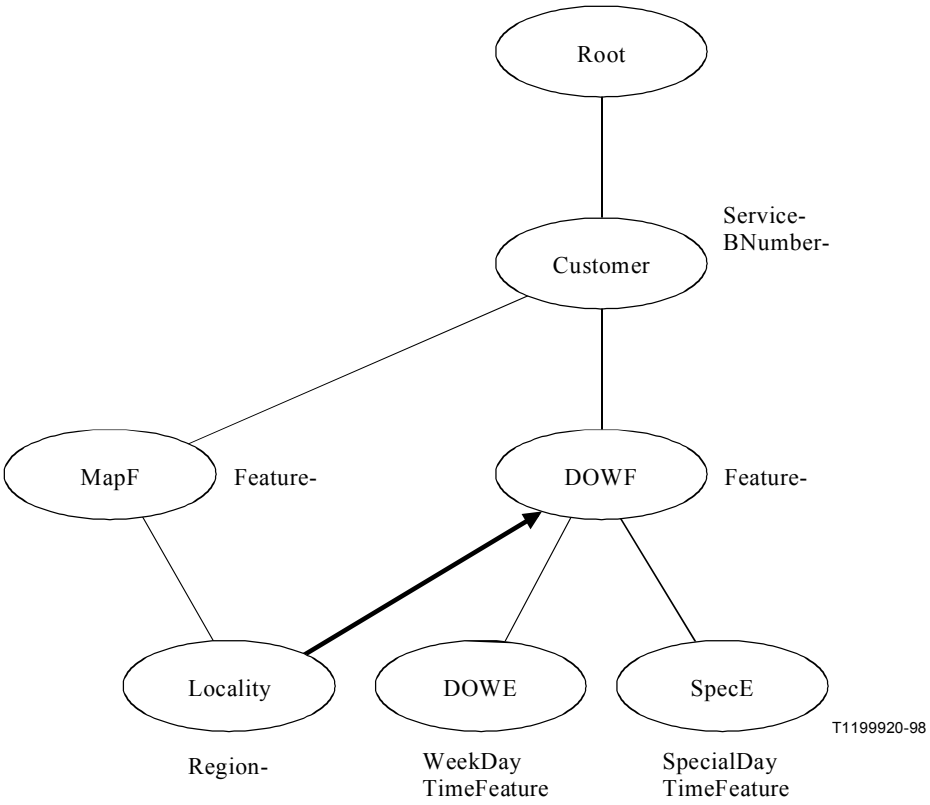


Figure 7-25/Q.1229 – Information model used to select the day-of-week

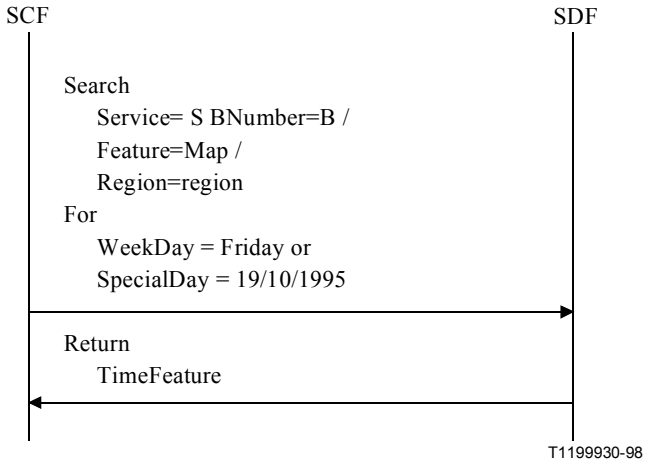


Figure 7-26/Q.1229 – Information flows used to select the day-of-week

The translation from the time-of-day to a splay feature is a straightforward comparison between the current time and a list of time-of-day entries. As with the day-of-week list, the time-of-day list could use multi-valued attributes with application contexts instead of a list of subordinate entries. Figure 7-27 shows in more detail the schema used in this operation. Figure 7-28 shows the information flows used to implement the operation.

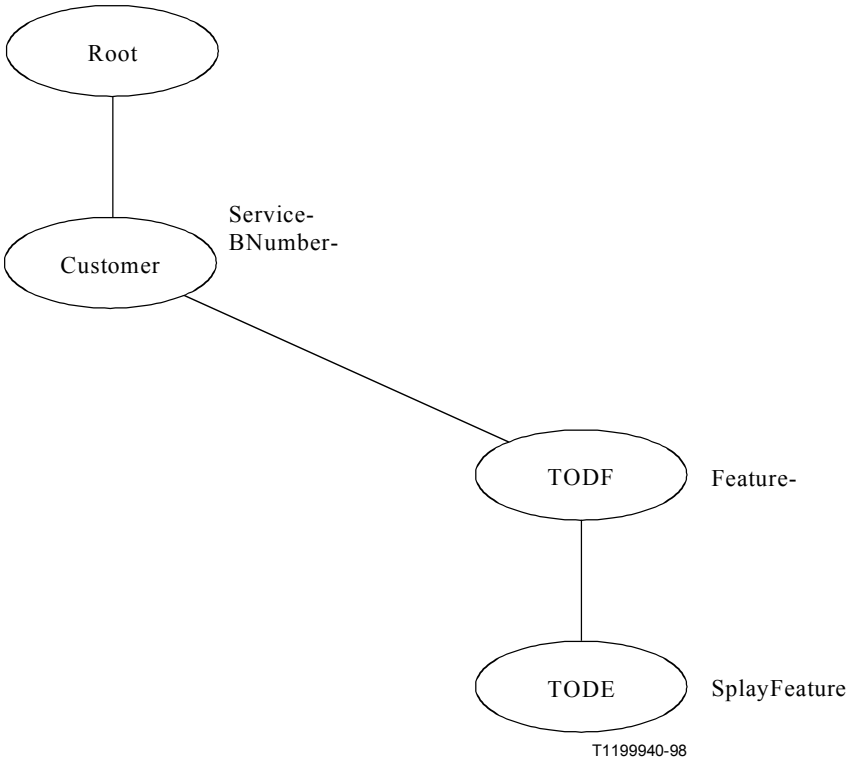


Figure 7-27/Q.1229 – Information model used to translate the time to a splay feature

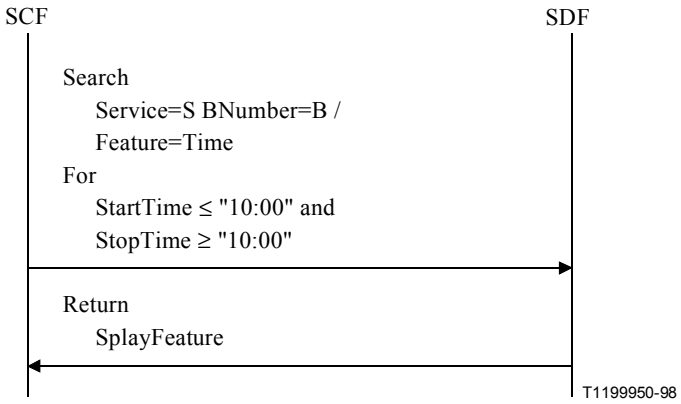


Figure 7-28/Q.1229 – Information flows used to used to translate the time to a splay feature

In this implementation, calls are distributed based upon the number of calls using the selected splay entry. Each splay entry has a fixed number of splay destinations. For each destination it lists a required percentage of the number of calls to be made to that destination, the name of the C-number entry (CNumF) that the calls are to be distributed to, and actual number of calls that have been distributed to that entry. To select the appropriate destination, the SCF must retrieve all of the

splaying information and execute the splaying algorithm (in the SCF). Once this has been done, the number of calls to that destination is incremented. The "TranslatedNumber" and "NumberOnBusy" are then retrieved from the selected CNumF entry. Figure 7-29 shows in more detail the schema used in this operation. Figure 7-30 shows the information flows used to implement the operation.

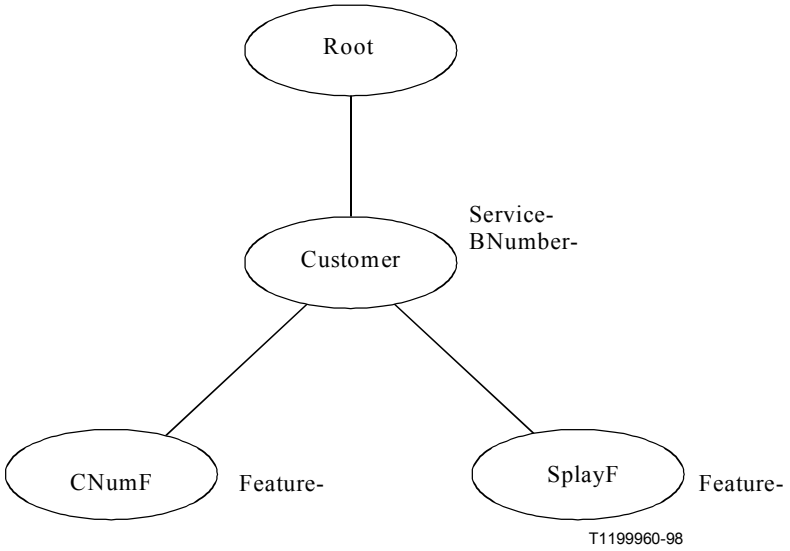


Figure 7-29/Q.1229 – Information model used to distribute calls based on percentage distribution

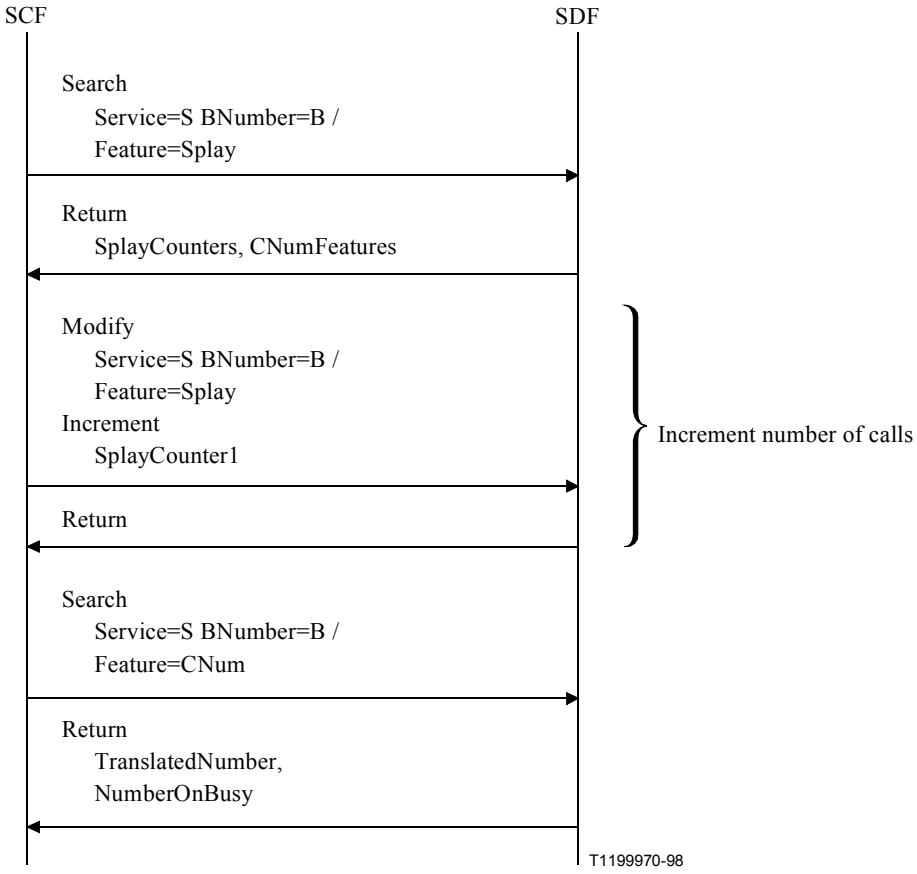


Figure 7-30/Q.1229 – Information flows used to used to distribute calls based on percentage distribution

The total number of database requests for this service, ignoring binds and unbinds, is six.

7.2.3.4.2.5 Using entry methods

The number of database operations required to implement a service with reasonably complex data manipulations can become excessively large (refer to the above subclause). If, however, entry methods are used, the number of external database operations falls to just one while still retaining all of the data complexity. Figure 7-31 shows the data model, as seen external to the SDF, for the example flexible call routing service.

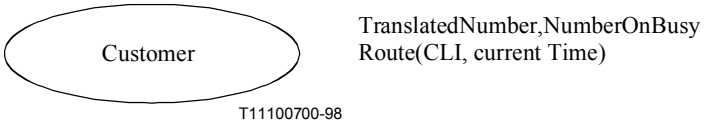


Figure 7-31/Q.1229 – Alternative information model

This can be described using the following ASN.1 notation:

```
SupportedMethods METHOD ::= { Route | ... }

Route METHOD ::= {
        INPUT ATTRIBUTE CLIInfo
        OUTPUT ATTRIBUTE SelectedNumbers
        ID route-opcode
}
CLIInfo ::= SEQUENCE {
        cli-prefix              DigitString
  current-time       DateAndTime
}
SelectedNumbers ::= SEQUENCE {
        TranslatedNumber        DigitString
        NumberOnBusy            DigitString
}
```

In this example the "Customer" object-class has a method named "Route" which includes all of the operations shown in the previous implementation. This method takes as its input a CLI and the current time [currentTime]. It then returns a TranslatedNumber and NumberOnBusy. Figure 7-32 shows the single SCF-SDF information flow used to execute the method for an instance of Customer in the directory information tree where the name of the entry is "Service=S BNumber=B".

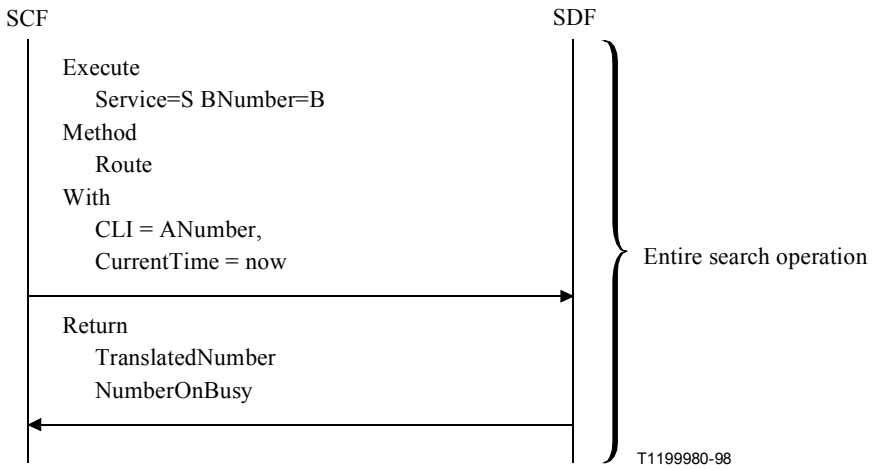


Figure 7-32/Q.1229 – Information flows used to implement sample service

The total number of database requests for this service, ignoring binds and unbinds, is one.

7.2.3.5 Generic METHOD to assign a unique value of a pool of resources

7.2.3.5.1 Background

It is common in telecommunications to have a protocol to access to a resource distinct from the protocol to use to operate it. The protocol to access a resource is often a reservation. Sometimes, the reservation could be based on the state of a network or of a service. But in a number of situations, the reservation is based on a simple stated allocated/non-allocated. In these cases, the reservation could be done by linking two data (resource identity, user identity).

In principle, an idle resource could be taken only by one user and sometimes, for security, the selected resource has to be unpredictable (e.g. random selection).

Then, in Intelligent Network, there is a need to get an atomic mechanism to allocate (uniquely and temporarily) a resource (number/identity) to a user/terminal.

There are two well-known such situations with mobile services:

– allocation of a temporary identity at location updating (or at each call set-up) to preserve the anonymity of the user/terminal;

– allocation of a roaming number on a per call basis, which can serve both as a unique user identity and as a routing address to the current location.

7.2.3.5.2 Solutions

A reservation procedure is divided in two stages:

– Make the reservation.

– Free the reservation.

The first stage could be seen as two steps:

– Find an idle value.

– Link this value to the user.

There are several solutions.

1) *Reservation done by the SCF*

A solution could have been to do the procedures of selection-assignment/find-release in the Service Logic Program Instance in the SCF and stored the assigned values in the SDF. But this solution has many drawbacks:

a) The SCF manages data that it does not own.

b) The data related to the reservation associate a resource to a user during a period of time which goes beyond the duration of a Service Logic Program Instance.

c) The Information Flows which seem simple become very complex when the error cases and the concurrency of the allocations are taken into account.

d) The consumed bandwidth to correctly achieve this procedure from a SCF is not to be neglected.

e) In term of Service Logic, the SCF is only concerned to get the allocated values with the necessary properties (e.g. unicity, range). A value by itself like the memory address where it is stored is meaningless for the SCF; that the selected value was "123" or "321" has no impact on the SL.

f) The SCFs which need allocated data are generally in other networks but have to go by a unique SCF of the network which will provide the resource

2) *Reservation done by the SDF*

With a METHOD, a SDF is able to run a data reservation script to manage the allocation of a resource number to a user identity. This solution has many advantages:

a) The SDF manages its own data.

b) The SDF is there to manage SCF data which are used during a period of time which goes beyond the duration of a Service Logic Program Instance.

c) The atomic execute operation permits to get simple Information Flows.

d) The Information Flows are limited to two (request, result).

e) The script is independent of the Service Logic.

f) The SCFs can directly make their request of reservation to the SDF.

7.2.3.5.3 Object definition

Similarly to object-oriented-database modelling, a base object which will support the method must be defined. During processing, the method would be invoked on an instance derived from this base object or a child object.

The following object class is used to represent information related to the reservation procedures.

```
GenericAllocationPool {ATTRIBUTE assignmentTable
                    ,OBJECT IDENTIFIER: code} OBJECT-CLASS ::={
KIND        auxiliary
MUST CONTAIN {assignmentTable}
MAY CONTAIN {maxtime|randomAssigned}
ID code}
```

The GenericAllocationPool could be associated with an Organization Unit (or a subclass) OBJECT-CLASS to create an entry.

The **assignmentTable** provided as an ASN.1 CLASS parameter is a multi-valued attribute which supports two contexts:

– temporal context;

– assignmentContext.

It could be any attribute of an entry using the GenericAllocationPool auxiliary objectClass.

The **maxtime** attribute indicates the duration of time the reservation could be maintained. It is used to create a suitable temporal context value to be associated for a selected value.

```
maxtime ATTRIBUTE ::= {
WITH SYNTAX INTEGER
SINGLE VALUE   TRUE
ID id-at-maxtime}
```

The **randomAssigned** attribute indicates that the values of the assignmentTable attribute must be selected at random.

```
randomAssigned ATTRIBUTE ::= {
WITH SYNTAX BOOLEAN
SINGLE VALUE TRUE
ID     id-at-randomAssigned}
```

7.2.3.5.4 Methods definition

Reservation methods are two generic methods which permit reservation of a unique value in a pool. (one method to assign, the other to release).

```
selectAndAssign METHOD
 ::={
SPECIFIC-INPUT  DistinguishedName
                                -- The DN of the user to which
                                -- the selected value is temporarily assigned
OUTPUT ATTRIBUTES          assignmentTable
BEHAVIOUR "This method performs the following actions on the entry identified by the execute argument:
1)      Selects a value of the assignmentTable attribute which is not associated with a context or which is
        associated with an expired temporal context.
2)      Adds an assignmentContextValue equal to the specific input to the selected value.
3)      Adds a temporal context value so that the selected value becomes irrelevant after maxtime units of time.
4)      return the selected value without context values.
"
ID                          id-mt-selectAndAssign
}

findAndRelease METHOD ::={
INPUT ATTRIBUTE       assignmentTable
SPECIFIC-OUTPUT       DistinguishedName -- The DN of the user to which
                                -- the selected value was
                                -- temporarily assigned
BEHAVIOUR "This method performs the following actions on the entry identified by the execute argument:
1)      Find the value of the assignmentTable attribute which is equal to the one received in the input-assertions
        element of the execute argument.
2)      Remove from the DIB all its associated context values.
3)      If this value was associated with valid temporal context values and an assignmentContext Value, return the
        associated assignmentContext value (user DN)."
ID      id-mt-findAndRelease}
```

These generic METHODS could be used easily for supporting roaming number allocation procedures as follows:

```
-- example for the Roaming number
roamingNumberPool OBJECT-CLASS ::= GenericAllocationPool { ATTRIBUTE: roamingTable, OBJECT
IDENTIFIER:id-oc-roamingNumberPool}
```

```
roamingTable ATTRIBUTE {
WITH SYNTAX    NumericString (SIZE(1..ub-international-isdn-number))
ID             id-at-assignmentTable}

roamingNumberRule METHOD-USE-RULE ::= {
OBJECT CLASS TYPE id-oc-roamingNumberPool
MANDATORY METHODS {findAndRelease|selectAndAssign}}
```

7.2.3.6 Security supported by the SDF

7.2.3.6.1 Background

The security facilities described in 7.2.2.4.1.1 are required to offer security features to IN services.

7.2.3.6.2 Requirements

The keys of user/terminal must be stored with the user information. They must be protected against disclosure and tampering.

The following capabilities shall be offered:

a) The credentials of a user/terminal provided in a bind argument shall be verified before opening a dialogue.

b) The network (SDF) may be requested to authenticate itself back in the bind result.

c) The origin (user/terminal) of a message may be authenticated (one-pass).

d) The user may use several keys (e.g. a temporary key for the subscription, his key, his PIN, ...).

e) Several algorithms may be used to authenticate the user/terminal.

f) The keys and cryptographic algorithms may be used to make disposable tokens as described in the following subclause.

g) For an authentication based on a PIN, the access shall disable after several consecutive failures.

h) For anonymous services, credentials produced by a smart card shall be checked on a dialogue open between two network operators.

7.2.3.6.3 Object definition

Since the same object could be used to store compute or check the credential, the following object class could be used to store the necessary information about the user security (parameters and policy).

In case of UPT, for each UPT user, an entry of the **securityUserInfo** object class may be created, subordinate to each entry of class **uptUser**.

```
securityUserInfo OBJECT-CLASS ::={
MUST CONTAIN  {securityFacilityId|
                        secretKey|
                        identifierList}
MAY CONTAIN             {bindLevelIfOK|
                        currentList|
                        failureCounter|
                        lockSession
                        maxAttempts}
ID                      id-oc-securityUserInfo }
```

securityFacilityId is an attribute to name the verification (requirement c)

```
securityFacilityId ATTRIBUTE ::= {
        WITH SYNTAX
        SF-CODE
        EQUALITY MATCHING RULE      objectIdentifierMatch
        SINGLE VALUE                TRUE
        ID                          id-at-securityFacilityId}
SF-Code ::= OBJECT IDENTIFIER
```

The **securityFacilityId** could have different values:

– id-sf-pwd for management access to the database by password

– id-sf-challengeResponse for standard access based on one pass challenge response authentication

– id-sf-onAirSubscription to authenticate the access during on-air subscription (the entry contains the same **identifierList** that the previous entry but **secretKey** is different.

```
-- Security Facility id
id-sf-pwd SF-CODE ::= {id-sf pwd(1)}
id-sf-challengeResponse SF-CODE ::= {id-sf common (2)}
id-sf-onAirSubscription SF-CODE ::= {id-sf subscription(3)}
```

secretKey is an attribute which contains the secret key (to be used by the cryptographic algorithm) of the user.

```
secretKey ATTRIBUTE ::= {
        WITH SYNTAX     BIT STRING (SIZE(lb-secretKey..ub-secretKey))
        SINGLE VALUE    TRUE
        ID              id-at-secretKey}

-- The following values are merely examples
lb-secretKey INTEGER ::= 32 -- the boundary values could be expanded
ub-secretKey INTEGER ::= 128 -- by a network operator
```

identifierList is an attribute which could contain four identifiers (requirement d):

– **conformMethodIdentifier** identifies the method used to verify that some parts of the input message conform to specified criteria such as size, value matching with an attribute, greater than a counter, included in a time window (requirement b[1]),

– **fillMethodIdentifier** identifies the method use to fill the input message (first part of a **twoPartMessage** or **ThreePartMessage or FivePartMessage)** (requirement e).

– **oneToOneAlgorithm** (and respectively **oneToTwoAlgorithm**) identifies the cryptographic algorithm with one output (respectively two output).

– If KS is the secret key, IN is the input and OUT the output, A1 and A2 cryptographic algorithms, it would be OUT = output1of (A2(RS_size_in_bits first bits of IN,A1(RAND_size_in_bits last bits of IN,KS))) (respectively (OUT1,OUT2)= (A2(RS_size_in_bits first bits of IN,A1(RAND size in bits last bits of IN,KS))).

```
identifierList ATTRIBUTE ::=
{
WITH SYNTAX
SEQUENCE{
conformMethodIdentifier [1]     MethodIdentifier, -- e.g. time window check
fillMethodIdentifier    [2]     MethodIdentifier-- e.g. generate a random of required size,
oneToOneAlgorithm       [3]     AlgorithmIdentifier  -- e.g. A11 and A12, output RES from RS,RAND
oneToTwoAlgorithm       [4]     AlgorithmIdentifier } -- e.g. DECT algorithm output RES,SDK from RS,RAND
```

---

[1] It is common in security (e.g. UPT authentication by DTMF, ECMA GSS-API) when the verifier draws the challenge value to check that it is not a replay of a previous value. Two mechanisms could be used: a counter or a concatenation of the current time window and a random.

```
SINGLE VALUE
        TRUE
ID                      id-at-identifierList}

-- AlgorithmIdentifier could be imported from ITU-T Rec. X.509
AlgorithmIdentifier ::= SEQUENCE {
algorithm     ALGORITHM.&id ({SupportedAlgorithms}),
parameters   ALGORITHM.&Type({SupportedAlgorithms}{@algorithm}) OPTIONAL}
MethodIdentifier ::= SEQUENCE {
methodid                    METHOD.&id ({SupportedMethods}),
inputAttributes             METHOD.&InputAttributes ({SupportedMethods}{@method}) OPTIONAL,
specific-Input              METHOD.&SpecificInput ({SupportedMethods}{@method}) OPTIONAL}
```

bindLevelIfOK is a mono-valued attribute which contains an **AuthenticationLevel**. It is to be used by the bind operation with the argument of the abstract-syntax defined in 7.3/Q.1228 to determine the level of privileges granted to the user. When this attribute is absent and a bind operation is invoked, the bind operation returns an error (requirement a).

```
bindLevelIfOK ATTRIBUTE ::=
{
WITH SYNTAX
AuthenticationLevel
SINGLE VALUE    TRUE
ID      id-at-bindLevelIfOK}
```

lockSession is a mono-valued attribute that contains the name of the entry and the mono-valued attribute of type boolean of this entry used to lock a dialogue to a mono-session (the timer set as temporal context on this lock attribute is the same for all the users). If this attribute is present and a bind operation is at the origin of the method invocation, the method first checks that the pointed attribute is FALSE before proceeding.

This optional attribute could be used in prepaid service or VCC when to avoid fraud (bypassing of the user credit) no concurrent session is accepted for this account.

```
lockSession ATTRIBUTE ::= {
WITH SYNTAX             LockSession
SINGLE VALUE            TRUE
ID              id-at-lockSession}

LockSession  ::= SEQUENCE {
entryName           [0]     DistinguishedName,
attribute           [1]     ObjectIdentifier}
}
```

For some security facilities, it is useful to count the number of failures and if necessary to lock the facility when a threshold is reached. The following two attributes are used to store these information (requirement f).

```
failureCounter ATTRIBUTE ::= {
WITH SYNTAX                             INTEGER
ORDERING MATCHING RULE                  integerOrderingMatch
SINGLE VALUE                            TRUE
ID                                      id-at-failureCounter}

maxAttempts ATTRIBUTE ::= {
WITH SYNTAX                             INTEGER
ORDERING MATCHING RULE                  integerOrderingMatch
SINGLE VALUE                            TRUE
ID                                      id-at-maxAttempts}
```

To control that no replay is done with the challenges RAND already drawn, it is necessary to maintain a list of the randoms already used for the valid period indicated by RS. The currentList attribute contains a list of RAND already used for the current period of time (requirement b).

```
currentList ATTRIBUTE ::= {
        WITH SYNTAX                     BIT STRING,
        EQUALITY MATCHING RULE          bitStringMatch
        ID                              id-at-currentList}
```

7.2.3.6.4 Methods definition

The METHOD verifies the user credential against the information included in an entry of type **securityUserInfo**. This METHOD could be used during the bind or over a dialogue to authenticate the user in the database. It could be used for example when the user changes service data over a management access.

```
verifyCredentials METHOD
::={
        SPECIFIC-INPUT  TwoPartMessage
        -- see the definition of this type below
SPECIFIC-OUTPUT
BOOLEAN
        -- to indicate the success of the verification
BEHAVIOUR "This method performs the following actions on the entry identified by the execute argument; this
entry would be of class genericSecurityUserInfo:
1)      if maxattempts is present, verify that failureCounter is less than its value
2)      read the value of identifierList attribute  (return "bad format entry" if failure)
3)      if conformMethodIdentifier is NULL go to step 5)
4)      run conformMethodIdentifier method on TwoPartMessage provided as specific input (return a
        "badconformance" error if the execution fails or if the result is false)
5)      run the oneToOneAlgorithm on the messageData bit string to get an expected certificationCode bit string
6)      return TRUE if the expected and provided certificationCode values match and exit,
7)      otherwise if failureCounter is present, increment it and return FALSE
"
ID                              id-mt-verifyCredentials}
```

The value of **conformMethodIdentifier** could be **id-mt-conformCredentials**.

```
ConformCredentials METHOD ::={
SPECIFIC-INPUT  TwoPartMessage
        -- see the definition of this type below
SPECIFIC-OUTPUT         BOOLEAN
        -- to indicated the success of the verification
BEHAVIOUR "This method performs the following actions on the entry identified by the execute argument; this
entry would be of class genericSecurityUserInfo:
–       verify with an embedded conformance algorithm that messageData value of TwoPartMessage is no replay
        (RAND is in the current time window and the associated RS is not in the list of the current time windows
        currentList).
–       add RAND to time windows list currentList.
–       return TRUE if no replay,
–       otherwise return FALSE
"
ID                              id-mt-dectConformCredentials}
```

The object class **SecurityUserInfo** supports the method **verifyCredentials.**

```
securityUserInfoRule METHOD-USE-RULE ::= {
OBJECT CLASS TYPE           id-oc-securityUserInfo
MANDATORY METHODS           {verifyCredentials| fillSecurityTokens|conformCrendentials}}
```

In the case of a visited network, entry (of **objectClass challengeResponseStock**) will contain in the SDF visited the DN of the entry (of **objectClass challengeResponseStock**), and will contain in the SDF home the DN of the entry (of **objectClass securityUserInfo**).

The entry **securityUserInfo** designated by the DN will contain in its **identifierList** attribute the value **id-mt-fillSecurityTokens** in the field **fillMethodIdentifier**.

```
NPARTMESSAGE{INTEGER : n} ::= SEQUENCE SIZE(2..n) OF BIT STRING

fillSecurityTokens {NPARTMESSAGE, OBJECT IDENTIFIER : code} METHOD ::={
SPECIFIC-INPUT  INTEGER -- X number of value to be computed
SPECIFIC-OUTPUT          SEQUENCE OF NPARTMESSAGE
BEHAVIOUR "This method performs the following actions on the entry identified by the execute argument, this
entry shall be of object class (or subclass) genericSecurityUserInfo:

–   read the secretKey attribute and Algorithms attribute
–   repeat X times
      –   fill the first BIT STRING field with a random value
      –   apply cryptographic algorithms to compute
          the other BIT STRING fields of the NPARTMESSAGE.
–   return X NPartMessage values
"
ID          code

-- id-mt-fillSecurityTokens-N
}
```

7.2.3.6.5 Security token

7.2.3.6.5.1 Background

In telecommunication mobile systems, it is current that a user is roaming in a visited domain far from his home domain; it could be expensive to dialogue each time with the home network if it is only to authenticate the user in the visited domain (e.g. simple call without modification of user service data). In systems as GSM in Europe, a stock of security tokens (challenge, response, ciphering session key) is provided to the visited domain in order to authenticate the user and optionally the cipher on the radio channel.

7.2.3.6.5.2 Object definition

This object class is used to represent a set of information which is common to all disposable tokens (stock identifier, source, size of the set). Disposable tokens could be, for example, authentication tokens pairs, triplets.

```
tokensStock {INTEGER: n, OBJECT IDENTIFIER : code } OBJECT-CLASS ::={
KIND                    abstract
MUST CONTAIN            {stockId | stock{n}}
MAY CONTAIN             {source| sizeOfRestocking}
ID                  code -- id-oc-tokensStock-n
}
```

stockId is a mono valued attribute of type DT-Code that is used as a naming attribute.

```
stockId ATTRIBUTE ::= {
      WITH SYNTAX                   DT-Code
      EQUALITY MATCHING RULE        objectIdentifierMatch
      SINGLE VALUE                  TRUE
      ID                            id-at-stockId}

DT-Code ::= OBJECT IDENTIFIER
```

source is a mono-valued attribute of type choice.

source ATTRIBUTE ::={
WITH SYNTAX SourceType
SINGLE VALUE TRUE
ID id-at-source}

SourceType ::=DistinguishedName

In the visited network, the **source** attribute will be used to store the DN of the entry of class derived from **stockId**. In the home network, the attribute will contain the DN of an entry of class **securityUserInfo** (defined in the previous subclause). The **token** is generated using the method defined in the **fillMethodIdentifier** field of this entry of class **securityUserInfo**.

sizeOfRestocking is a mono-valued attribute which indicates how many tokens have to be requested or computed when the **tokens** attribute is empty.

sizeOfRestocking ATTRIBUTE ::= {
WITH SYNTAX INTEGER
ORDERING MATCHING RULE integerOrderingMatch
SINGLE VALUE TRUE
ID id-at-sizeOfRestocking }

The following attribute could contain the precomputed set of (CHALLENGE,RES[,DCK][,NCHALLENGE,NRES]) (2, 3,4 or 5 values).

stock(INTEGER: n, OBJECT IDENTIFIER : code } ATTRIBUTE ::= {
WITH SYNTAX NPartsMessage{n}
ID code --*id-at-challengeResponse when n is two*
}

NPartsMessage{INTEGER : n} ::= SEQUENCE SIZE(2..n) OF BIT STRING

7.2.3.6.5.3 Methods definition

We consider that a DUA request to a DSA a stock of tokens by a method "provideTokens".

The role of DUA could be played by a SCF or by the local SDF to reprovision its stock.

The role of DSA could be played by the local SDF or by the home SDF.

provideTokens METHOD ::={
SPECIFIC-INPUT INTEGER, -- *how many tokens are requested (NofRT)*
OBJECT IDENTIFIER -- *oid of the attribute (tokens)*
SPECIFIC-OUTPUT ATTRIBUTE --*attribute selected as input (tokens)*
BEHAVIOUR "This method performs the following actions on the entry (thisEntry would be a variable with the DN value of this entry) identified by the execute argument:

1) If the attribute sizeOfRestocking doesn't exist in the entry, define a variable MAXNTsizeOfRestocking.
2) Verify that NofRT is inferior or equal to MAXNT (return an "execute error" if NofRT value is superior to MAXNT).
3) Read the attribute of the entry which has the selected oid and count the number of values (0 if empty) and put the result in a variable N (return "execute error" if the attribute doesn't exist).
4) Read the source attribute in the entry (return "execute error" error if source does not exist).
5) If N is inferior to NofRT and the DN of source indicates an entry of class or subclass tokenStock:
5a) Bind anonymous with the DSA which contains the entry defined by the address field of source.
5b) Execute the method provideTokens on the entry with MAXNT as value of the specific-input.
5c) If none error is returned, modify the tokens attribute by adding the resulted values.
6) If N is inferior to NofRT and the DN of source indicates an entry of class or subclass securityUserInformation:
6a) Execute the method defined by fillMethodIdentifier field value on the entry defined by the DN with MAXNT as specific input.
6b) If none error is returned, modify the tokens attribute by adding the resulted values.
7) Read the tokens attribute.
8) Define a variable "toBeReturned" with NofRT values of tokens attribute and a variable "toBeKept" with remainder values.
9) Remove tokens attribute.

10) Modify tokens attribute by adding the "toBeKept" values.
11) Return the "toBeReturned" values.
"

ID id-mt-provideTokens}

7.2.3.7 Information flow diagrams for SDF Shadow Updates

This subclause defines the SDF-SDF Shadow Update operations in the form of information flow diagrams. These diagrams in this subclause show the different options for mapping the operations on the TC layer and do not include information flows for errors or for unbinding the association. The states on these diagrams show the states of the relevant SDF application entities, consistent with 14.4.2.1/Q.1228.

7.2.3.7.1 Copy supplier initiated call

For the case where the copy supplier initiates the dialogue, Figures 7-33 to 7-36 are applicable.

Figure 7-33 shows the case where the DSAShadowBind, CoordinateShadowUpdate and UpdateShadow operations are all sent in separate TC PDUs. This approach is the only approach defined in Recommendation X.525, but may not be suited to IN applications due to the performance requirements of IN. A more suitable approach is to send multiple sequential SDF operations in a single TC PDU as if the first operation had been successful; this is shown in Figure 7-34.

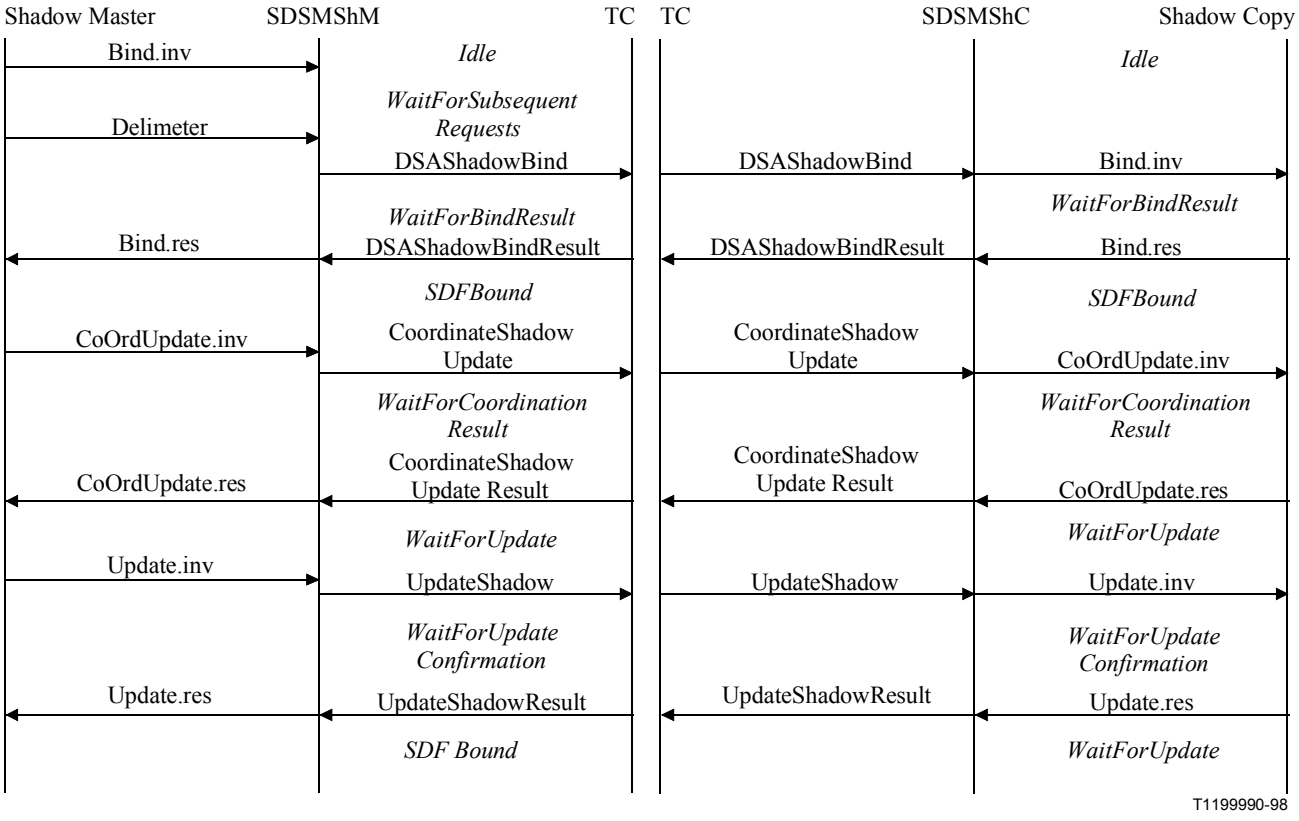


Figure 7-33/Q.1229 – Supplier initiated Copy Update using separate TC PDUs for each SDF-SDF operation

Figure 7-34 shows the case where the DSAShadowBind, CoordinateShadowUpdate and UpdateShadow operations are all sent in the same TC PDU. This is the preferred option, where efficiency is required and the shadow needs to be updated at the beginning of the dialogue.

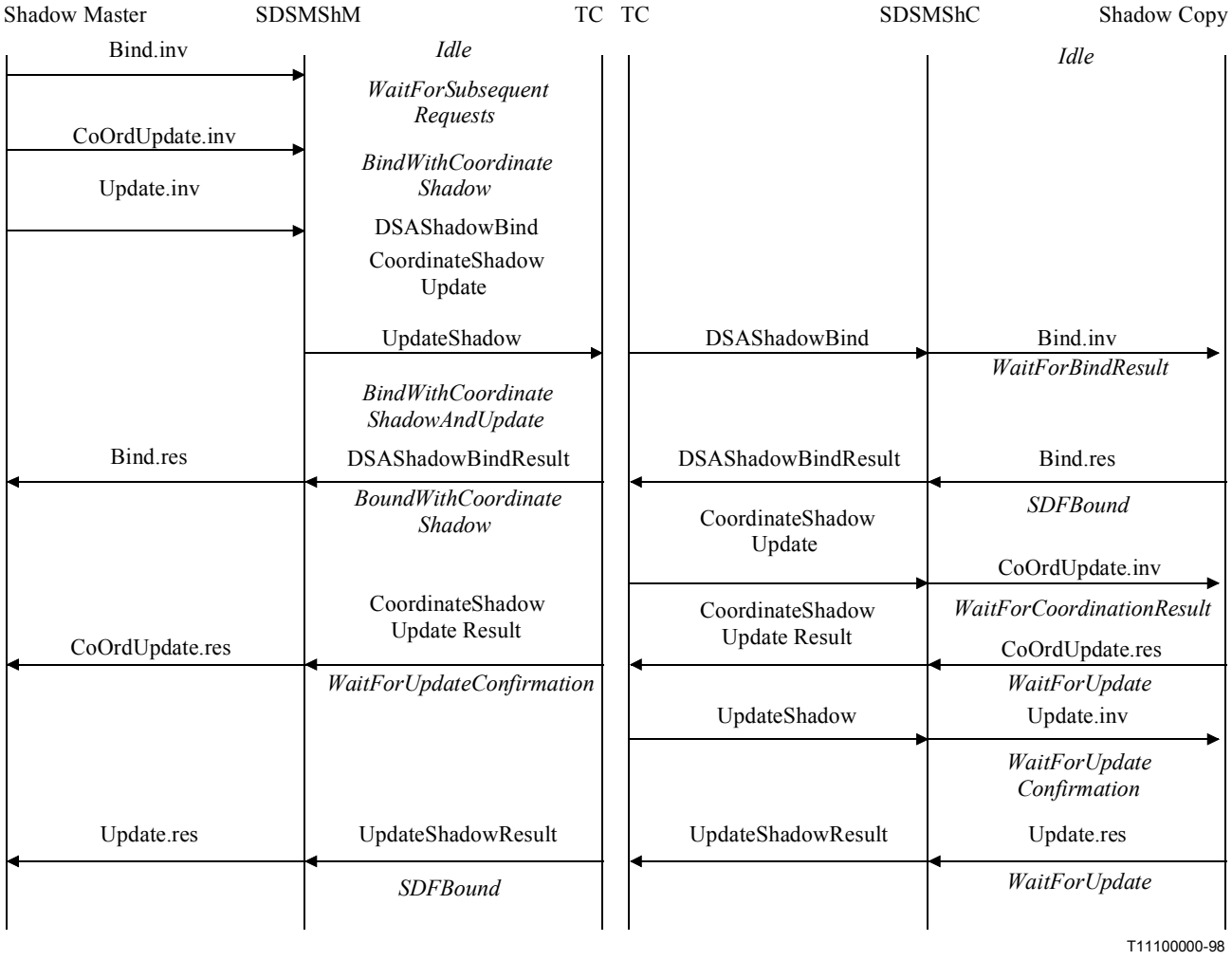


Figure 7-34/Q.1229 – Supplier initiated Copy Update using a single TC PDU on sending end

Additionally, the CopyConsumer may choose to bundle all the options on the returning end into a single TC message. It can be seen from Figure 7-35 that this does not affect the SDSMShC or SDSMShM state machines.

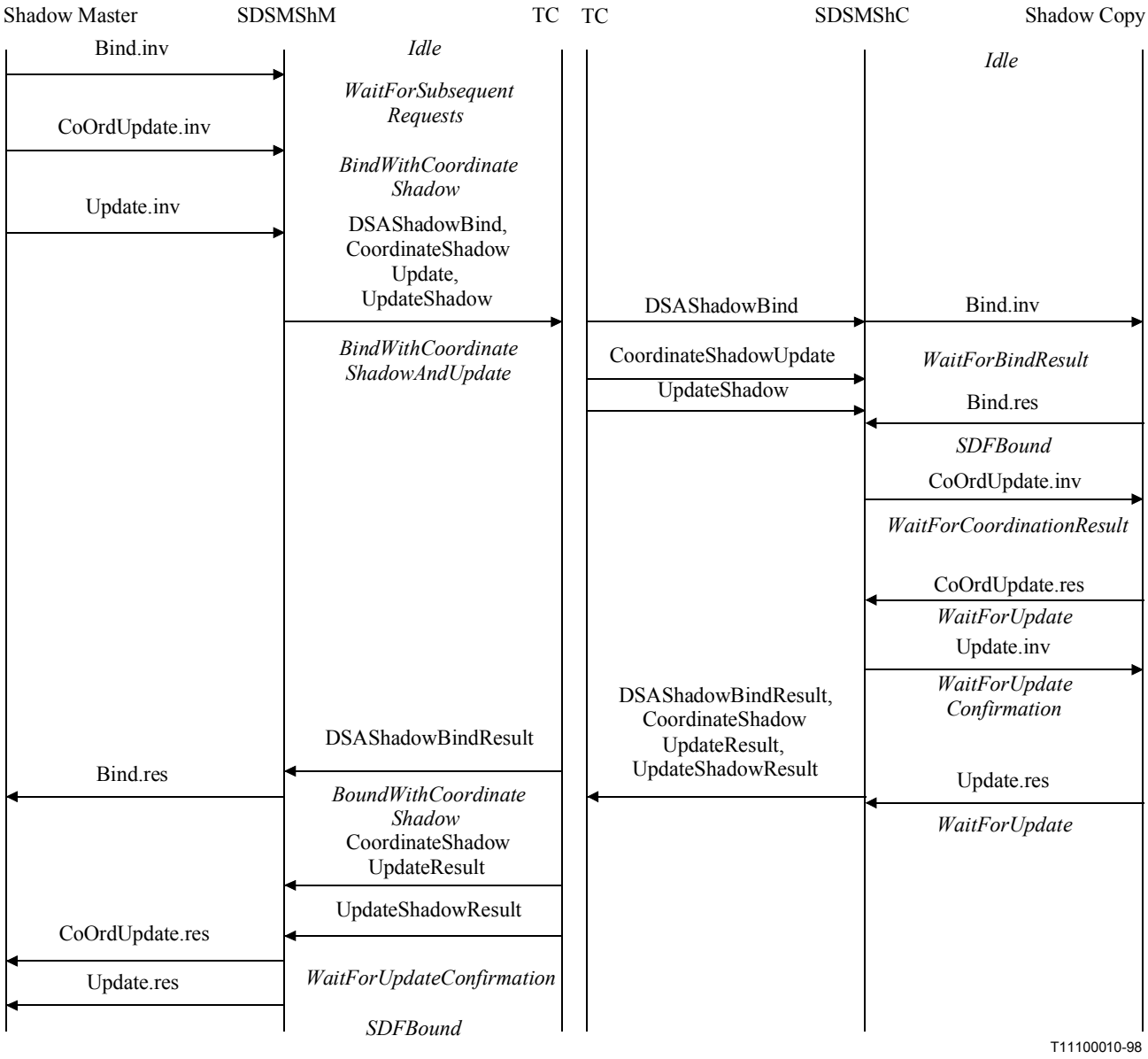


Figure 7-35/Q.1229 – Supplier initiated Copy Update using single TC PDUs on both sending and terminating ends

Figure 7-36 shows the case where the DSAShadowBind and CoordinateShadowUpdate operations are sent in the same TC PDU but the UpdateShadow operation is sent in its own TC PDU. This option is applicable where efficiency is required but the shadow supplier does not wish to update the shadow copy at the beginning of the dialogue.

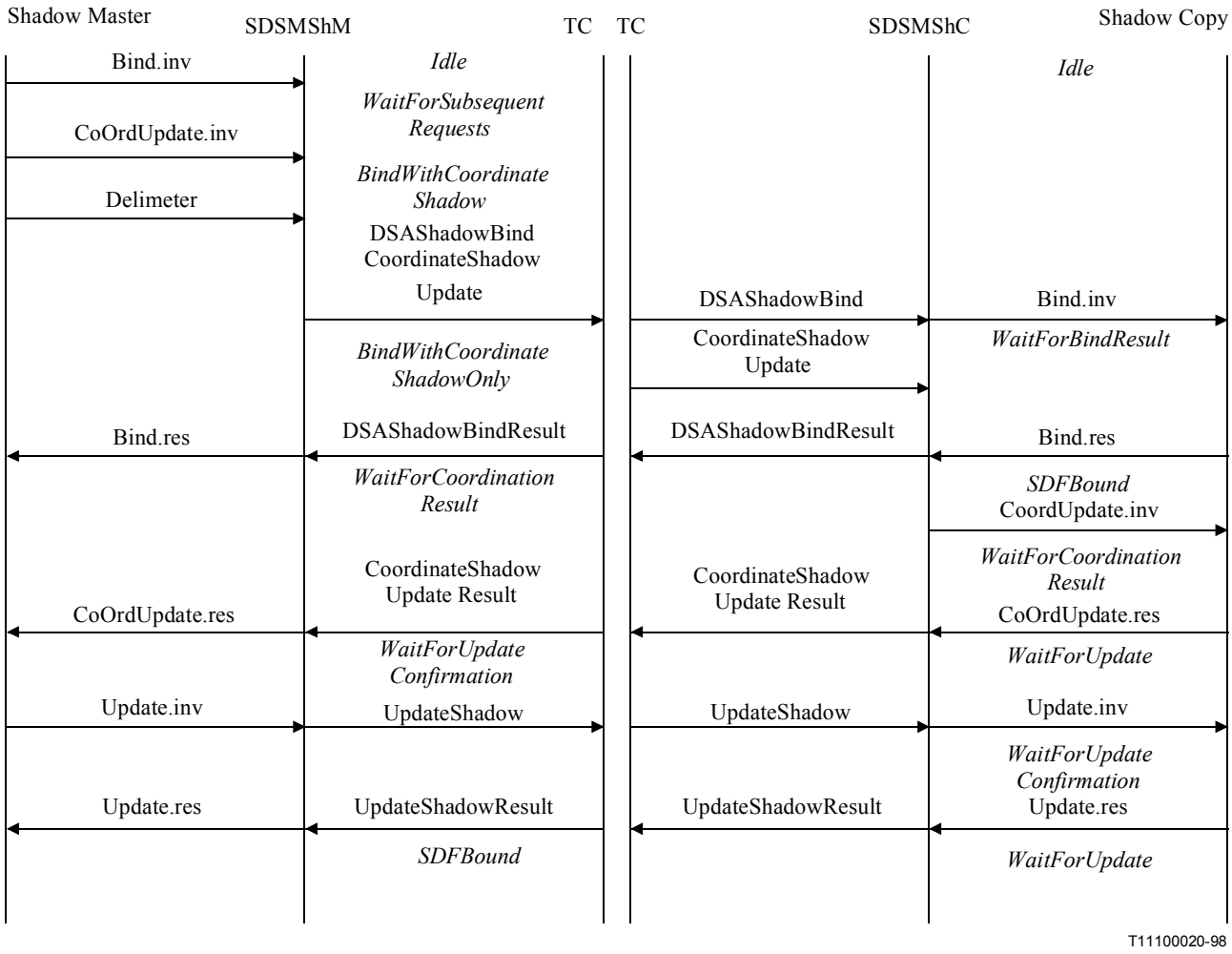


Figure 7-36/Q.1229 – Supplier initiated Copy Update using a single TC PDU on sending end for both Bind and Coordinate messages

7.2.3.7.2 Copy consumer initiated call

For the case where the copy consumer initiates the dialogue, Figures 7-37 to 7-39 are applicable.

Figure 7-37 shows the case where the DSAShadowBind, RequestShadowUpdate and UpdateShadow operations are all sent in separate TC PDUS. This approach is the only approach defined in Recommendation X.525, but may not be suited to IN applications due to the performance requirements of IN. A more suitable approach is to send multiple sequential SDF operations in a single TC PDUs as if the first operation had been successful; this is shown in Figure 7-38.

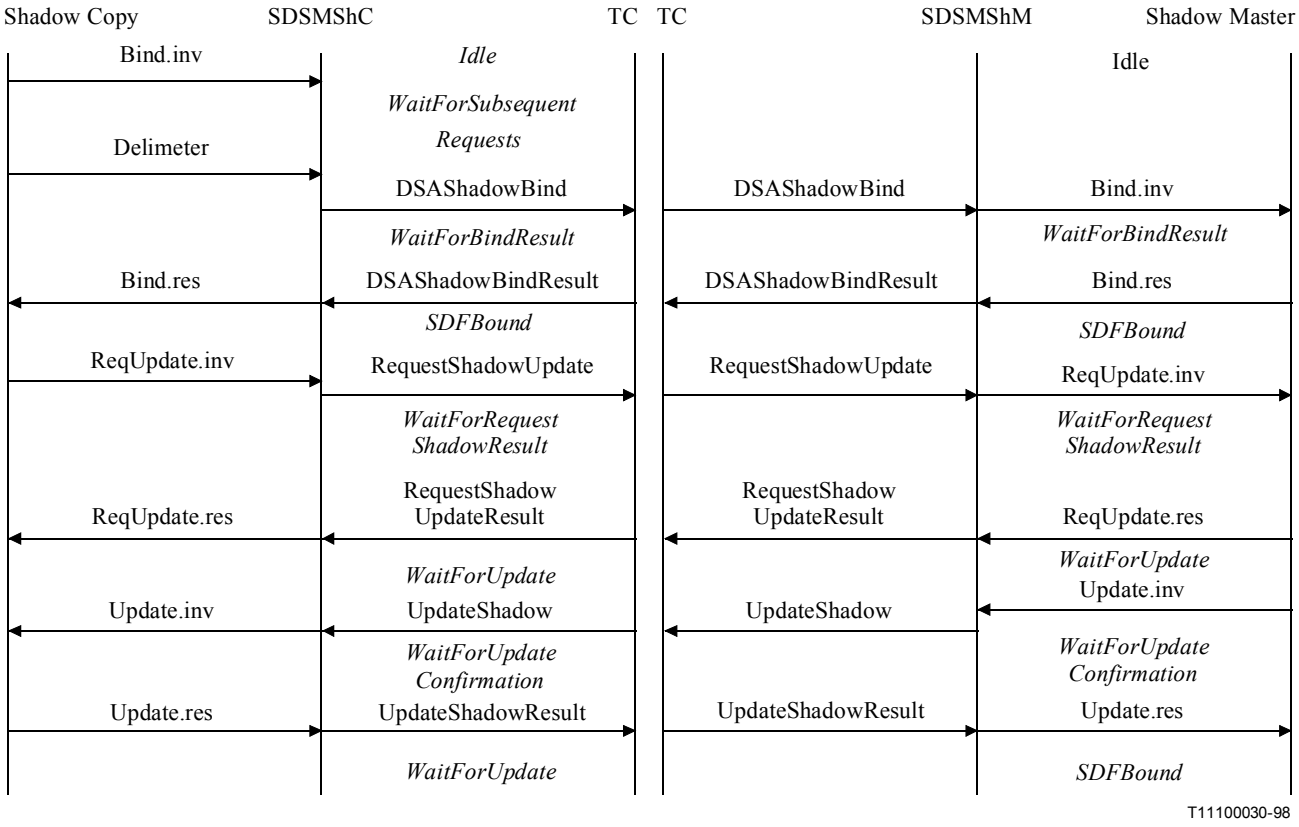


Figure 7-37/Q.1229 – Consumer initiated Copy Update using separate TC PDUs for each SDF-SDF operation

Figure 7-38 shows the case where the DSAShadowBind and RequestShadowUpdate operations are sent in the same TC PDU. Additionally, if the Shadow agreement involves updating the shadow at the initiation of the connection, the Copy Supplier TC may choose to bundle all the options on the returning end into a single TC message as shown in Figure 7-39.

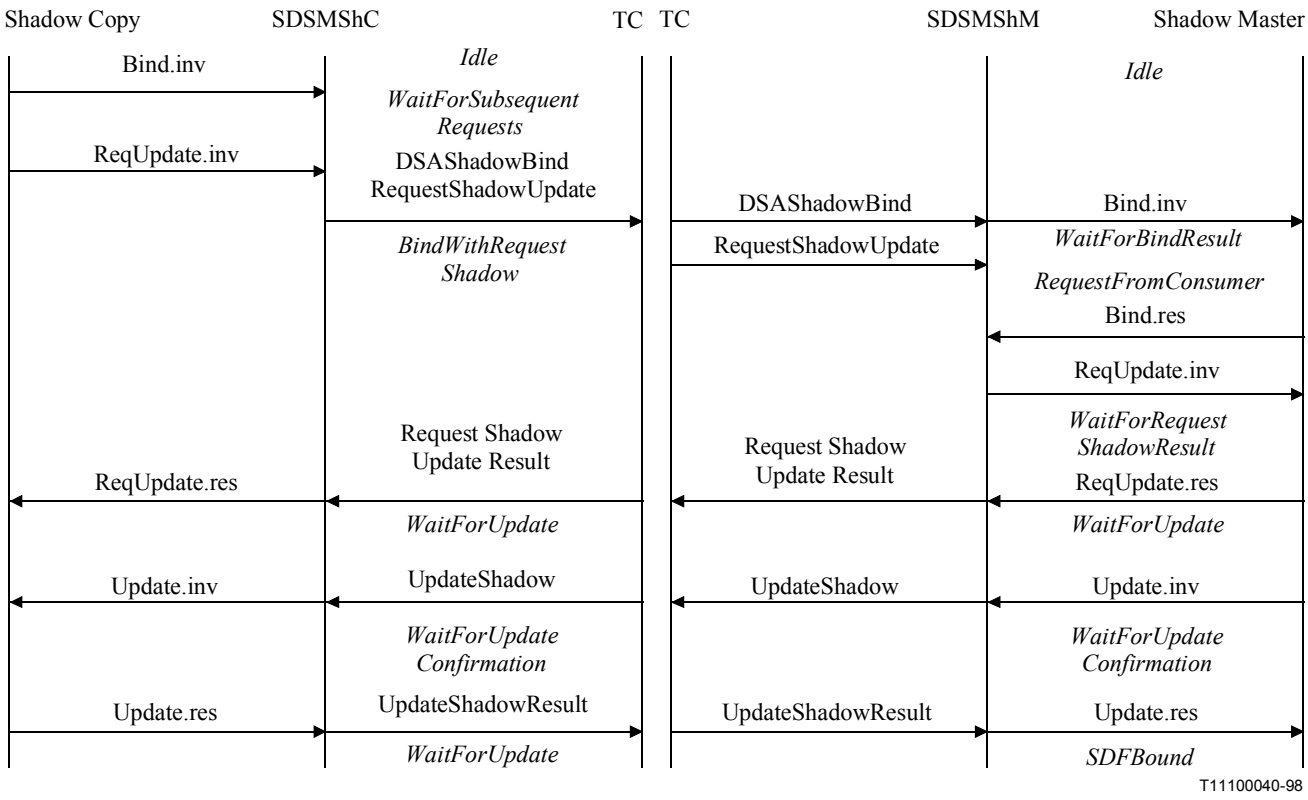


Figure 7-38/Q.1229 – Consumer initiated Copy Update using a single TC PDU on sending end

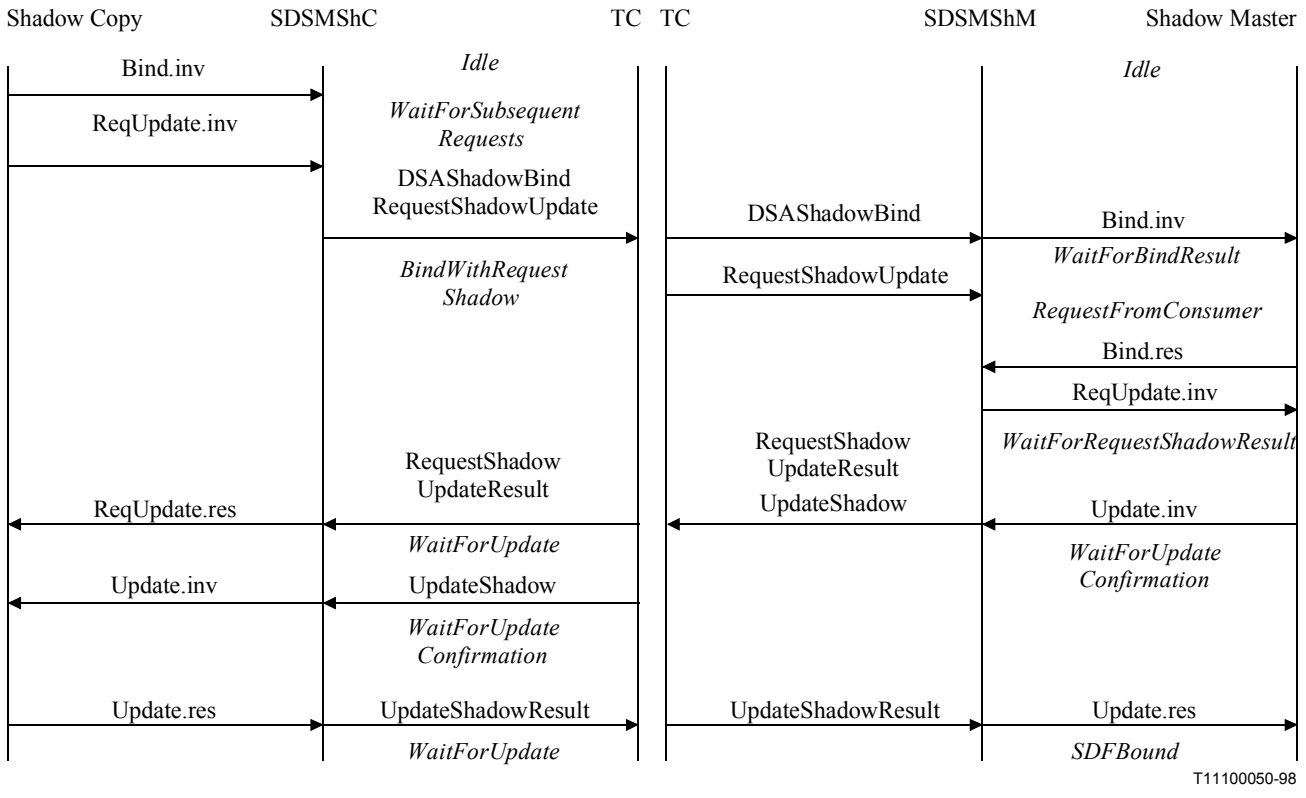


Figure 7-39/Q.1229 – Consumer initiated Copy Update using single TC PDUs on both sending and terminating ends

7.2.3.8 Mapping tables for signalling control primitives

7.2.3.8.1 Introduction

A SSF_CCF Basic Primitive Interface Model (BPIM) is provided in Annex A/Q.1228. That annex also includes the definition of the applied primitive signals in the INAP SDL model. The SSF_CCF Basic Primitive Interface Model (BPIM) allows to describe the applied signalling primitive interfaces and their possible mappings to applied signalling protocols The INAP SDL model consists of two half calls: one originating (SSF_CCF_A), the other terminating (SSF_CCF_B). In order to operate the model, both half calls are needed.

The BCSM is supposed to model existing switch processing of a basic two-party call and should reflect the functional separation between the originating and terminating portions of calls. The SSF-CCF BPIM includes a half call (SSF_CCF_A) with an Originating BCSM and a half call (SSF_CCF_B) with a Terminating BCSM. In this way the full functionality of the interworking between the O_BCSM and the T_BCSM is catered for. Since the BCSM is generic, it may describe events that do not apply to certain access arrangements (e.g. analogue signalling systems).

The SSF_CCF Generic Primitive Interface Model supports four different interface types: the SigCon interface to/from NNI/UNI [e.g. ISUP/DSS1, the IBI interface between half calls and the INAP interface to/from the standardized INAP messages (operations)].

The signalling control interface is a generic interface that can be mapped to different signalling protocols. Mappings examples are provided from SigCon_A respectively SigCon_B primitive signals to DSS1 and ISUP messages and shown in mapping tables for each half call. However, mapping to other signalling protocols may as well be applied.

Between the two call halves a switch internal intra BCSM Interface (IBI) carrying abstract generic primitive signals is applied.

The generic primitive signals used in the SDLs are aligned with the information flows described in Recommendation Q.71. The primitive signals defined supporting the UNI/NNI interface are attached to improve the readability of the mapping tables. The SCF-SSF primitives supporting the INAP interface are defined in Annex A/Q.1228 and are not listed here as they map directly to the corresponding INAP operations defined in clause 17/Q.1228.

7.2.3.8.2 Examples of mapping tables for IN used primitive signals

7.2.3.8.2.1 How to read the tables

The mapping tables provided are to show the generic primitive signals and their interfaces and proposed possible mapping to applicable agent signals. As an example ISUP and DSS1 are here indicated as possible applied agent protocols. However, any applicable agent protocol may be used. It may be possible to derive the actual specification for any agent protocol by using this description in combination with the appropriate interworking specification. In the tables, references are made to the primitive signalling interfaces (e.g. in forward direction the interfaces c, e and g) and to the agent signalling interfaces (e.g. interfaces a and i) as shown in the SSF/CCF Primitive Signal Interface model in Figure 7-40. Furthermore the term "influence" is applied in the tables to indicate where the SCF may have the ability to impact call signalling procedures, e.g. ISUP messages and parameters. Where a mapping to UNI/NNI appropriate signalling messages for IN CS-2 support is required and not known, this is indicated as to be determined (tdb).

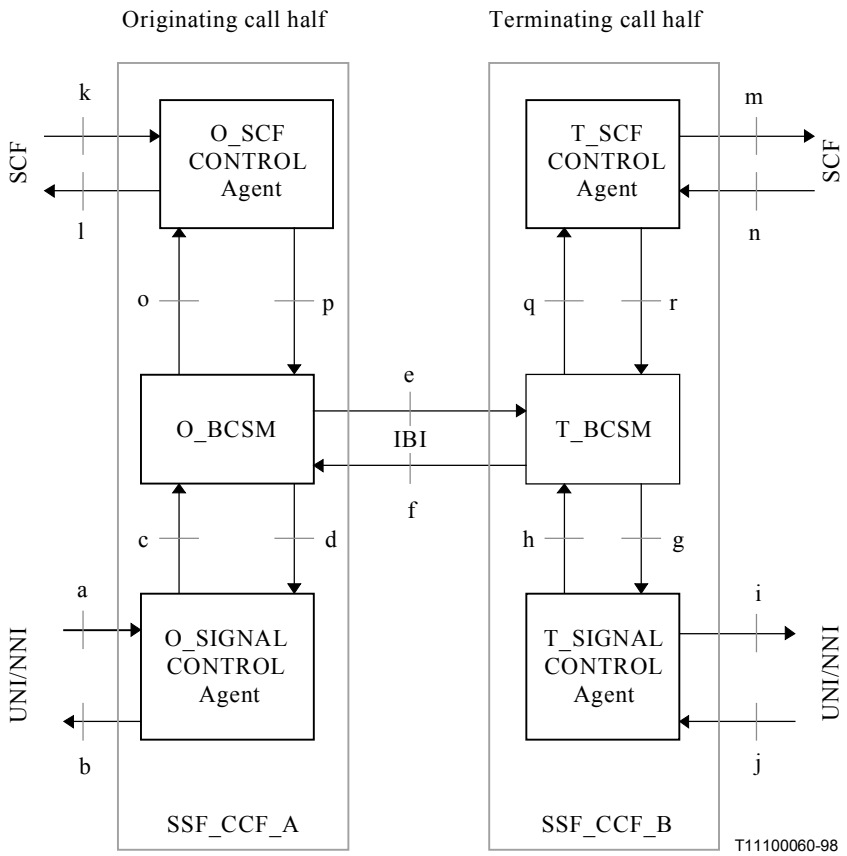


Figure 7-40/Q.1229 – SSF_CCF basic generic Primitive Signal Interface model

7.2.3.8.2.2 Primitive signal conventions

Each Primitive signal will as a mandatory parameter include a CallRef parameter consisting of a CallFlag and a CallID (call instance identifier). The Call Flag indicates the direction of the primitive signal as indicated in Figure 7-41.

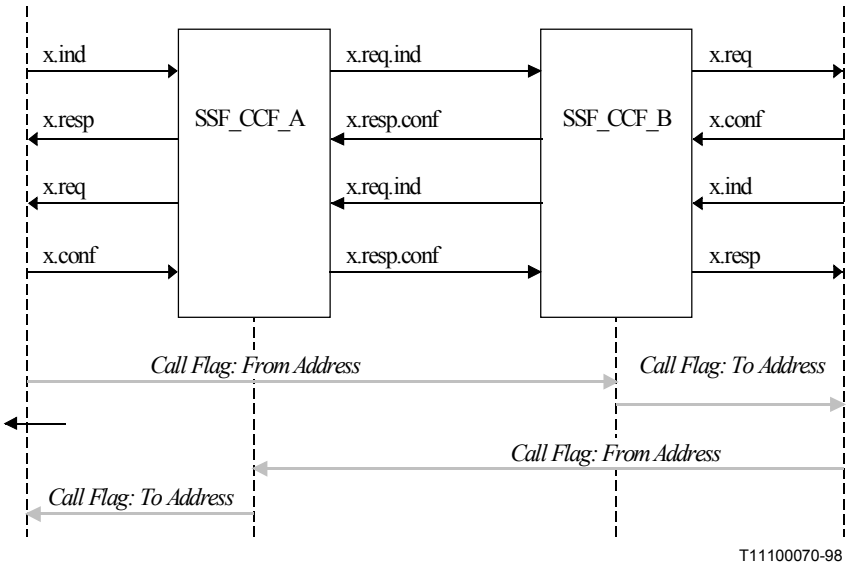


Figure 7-41/Q.1229 – Primitive conventions

Primitive signal types

- *Confirmed*

  Example: B party answer message received in response to a setup request message (e.g. ANM, CON in ISUP).

- *Unconfirmed*

  Example: B party alerted, Alert message sent backward to notify calling party (e.g. ACM (free subscr.) or CPG(Alert) in ISUP).

- *End-to-End*

  Example: Call setup messages that need end-to-end messaging. (e.g. IAM, ACM, ANM, CON in ISUP)

- *Link-by-Link (L)*

  Example: Release request from the network or a call party (e.g. REL/RLC in ISUP)

7.2.3.8.2.3 Primitive signal definitions

The generic primitive signals used in the SDLs are aligned with the information flows described in Recommendation Q.71. The primitive signals defined supporting the UNI/NNI interface are included here in order to ease the readability of the mapping tables. The SCF-SSF primitives supporting the INAP interface are defined in Annex A/Q.1228 and are not listed here as they map to corresponding INAP operations defined in clause 17/Q.1228.

7.2.3.8.2.4 Description of UNI/NNI related primitives

Setup

The Setup primitive is used to request establishment of a call connection. This is a confirmed signal, i.e. a response confirmation Setup primitive is used to confirm that the connection has been established.

The request for establishment of a connection can be originated by either the user or the network (i.e. SCF).

Release

The Release primitive is used to notify that a user has disconnected from the connection and cannot be connected and to request disconnection of a call connection. This is an unconfirmed signal.

SubsequentAddress

The SubsequentAddress primitive is a called number (address) signal for conveying subsequent address information during the digit-by-digit methods of call setup. and for conveying information about last digit received, i.e. address end during the digit-by-digit methods of call setup. This is an unconfirmed signal.

CallProgress

The CallProgress primitive is a signal that is used to report status and/or other types of call information across the network. The type of information is indicated (e.g. "no indication", "alerting", "remote call hold", etc.). This is an unconfirmed signal.

NetworkSuspend

The NetworkSuspend primitive is a signal used to suspend the call on behalf of the called party upon receipt of an on-hook indication from the terminating line or upon receipt of a network suspend message indication from terminating side. This is an unconfirmed signal.

NetworkResume

NetworkResume primitive is a signal used to resume the call on behalf of the called party upon receipt of a re-answer indication from the terminating line as the subscriber goes off-hook or upon receipt of a network resume message indication from terminating side. This is an unconfirmed signal.

Failure

Failure primitive is a signal used to report the occurrence of a failure in the network.

Reconnect

Reconnect primitive is a signal used to reconnect a controlling call party (leg) to the call. The call party is alerted (e.g. power ringing and/or display information) as the request is given for reconnection of the controlling call party to the call (with a held call party).

7.2.3.8.2.5 Primitive Signals Mapping Tables, Originating Half Call

a) *Call control primitive signals*

Table 7-17/Q.1229 – Primitive Signals, mapping to signalling Agent Protocols, Originating Half Call

Interface primitives	O_SIGNAL CONTROL (c/d)	IBI signal (e/f)	Information notes	Agent Protocol ISUP (a/b)	Agent Protocol DSS1 (a/b)
Setup	indication (c)	req.ind (e)	End-to-End	IAM	SETUP
Setup	response (d)	resp.conf (f)	End-to-End *) May be Discarded if backward resp. if sent previously	ANM, CON *)	CONNECT *)
SubsequentAddress	indication (c)	req.ind (e)	Link-by-Link May include both address digits(s) and indication for AddressEnd	SAM	INFORMATION
CallProgress	request (d)	req.ind (f)	End-to-End *)May be discarded if sent previously	ACM, CPG	ALERTING, PROGRESS
Release	request (d)	req.ind (f)	Link-by-Link B-Party (or SSF) initiated disconnect	REL/RLC	DISCONNECT
Release	indication (c)	req.ind (e)	Link-by-Link A-Party initiated disconnect	REL/RLC	DISCONNECT
NetworkSuspend	request (d)	req.ind (f)	End-to-End CS-2 *) "on-hook"	SUSPEND	- *)
NetworkResume	request (d)	req.ind (f)	End-to-End CS-2 *) "off-hook"	RESUME	- *)
ServiceFeature	request (d)	req.ind (f)	'Link-by-Link' B-party initiated Midcall event -- Applies to stimulus and functional terminal protocols		
ServiceFeature	indication (c)	req.ind (e)	A-party initiated Midcall event Applies to stimulus and functional terminal protocols	tbd	tbd

Table 7-17/Q.1229 – Primitive Signals, mapping to signalling Agent Protocols, Originating Half Call *(concluded)*

Interface primitives	O_SIGNAL CONTROL (c/d)	IBI signal (e/f)	Information notes	Agent Protocol ISUP (a/b)	Agent Protocol DSS1 (a/b)
Data	request (d)	req.ind (f) req.ind (e)	ServiceToUser-Information (Sent to A, received from T_BSM or sent to B)	tbd	tbd
Data	indication (c) request (d)	req.ind (e) req.ind (f)	UserToService-Information (send by A-Party or B-Party)	tbd	tbd
Failure	indication (c)	req.ind (e) req.ind (f) *)	*) Call release is initiated by SSF	REL/RLC	DISCONNECT
Reconnect	request (d)			tbd	tbd

b) SC-SSF primitive signals

Table 7-18/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Originating Half Call

Interface primitives	O_SCF CONTROL (o/p)	IBI signal (e/f)	O_SIGNAL CONTROL (c/d)	Information notes	Agent Protocols INAP (k/l)
ActivateService-Filtering	(p)	-	-	SSME SSF internal	ActivateService-Filtering
ActivityTest	(p)	-	-	*)SSF-SCF check	
AnalysedInformation	(o)	-	- Setup.ind (c) *)	(DP specific) *) DP report to SCF: routing addr. Available	AnalysedInformation -*)
AnalyseInformation	(p)	-influence (e) *) Setup.ind.req	-influence *)	(DP specific) *)- resumes originating basic call process.	AnalyseInformation
ApplyCharging	(p)	-	-	-	ApplyCharging

Table 7-18/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Originating Half Call *(continued)*

Interface primitives	O_SCF CONTROL (o/p)	IBI signal (e/f)	O_SIGNAL CONTROL (c/d)	Information notes	Agent Protocols INAP (k/l)
ApplyChargingReport	(o)	-	-	-	ApplyChargingReport
AssistRequest-Instructions	(o)	-	-		AssistRequest-Instructions
CallGap	(p)	-	-		CallGap
CallInformationReport	(o)	-	-		CallInformation-Report
CallInformationRequest	(p)	-	-		CallInformation-Request
Cancel	(p)	-	-	Cancel "all"	Cancel
CancelStatusReport-Request	(p)	-	-		CancelStatusRequest
CollectedInformation	(o)	-	-	(DP specific)	CollectedInformation
CollectInformation	(p)	-influence	influence		CollectInformation
Connect	(p)	influence	influence		Connect
ConnectToResource	(p)	-influence	-influence		ConnectToResource
Continue	(p)	-	-	resumes call processing	Continue
ContinueWithArgument	(p)	influence	influence	(INAP CS-2) resumes call processing	ContinueWith-Argument
CreateCSA	(p)	-	-	(INAP CS-2)	CreateCSA
DisconnectForward-Connection	(p)	-	- *)	- *) release of e.g. a temporary connection to an IP is not modelled	DisconnectForward-Connection
DisconnectLeg	(p)	influence	influence	(INAP CS-2)	DisconnectLeg

Table 7-18/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Originating Half Call *(continued)*

Interface primitives	O_SCF CONTROL (o/p)	IBI signal (e/f)	O_SIGNAL CONTROL (c/d)	Information notes	Agent Protocols INAP (k/l)
EntityReleased	(o)	-	-	(INAP CS-2)	EntityReleased
EstablishTemporary-Connection	(p)	- *)	- *)	*) set-up of a temporary connection to an IP is not modelled	EstablishTemporary-Connection
EventNotification-Charging	(o)	-	-		Event-Notification-Charging
EventReportBCSM	(o)	-	-		EventReport-BCSM
EventReportFacility	(o)	-	-	(INAP CS-2)	EventReport-Facility
FurnishCharging-Information	(p)	-	-		FurnishCharging-Information
HoldCallInNetwork	(p)	-	influence		HoldCallInNetwork
InitialDP	(o)	-	-		InitialDP
InitiateCallAttempt	(p)	influence	influence		InitiateCallAttempt
ManageTriggerData	(P	-	-	(INAP CS-2)	ManageTriggerData
MergeCallSegments	(p)	-	-	(INAP CS-2)	MergeCallSegments
MoveCallSegments	(p)	-	-	(INAP CS-2)	MoveCallSegments
MoveLeg	(p)	-	-	(INAP CS-2)	Moveleg
OAbandon	(o)	-	-	(INAP CS-2) (DP specific)	OAbandon
OAnswer	(o)	-	-	(DP specific)	OAnswer
OCalledPartyBusy	(o)	-	-	(DP specific)	OCalledPartyBusy
ODisconnect	(o)	-	-	(DP specific)	ODisconnect
OMidCall	(o)	-	-	(DP specific)	OMidCall
ONoAnswer	(o)	-	-	(DP specific)	ONoAnswer

Table 7-18/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Originating Half Call *(continued)*

Interface primitives	O_SCF CONTROL (o/p)	IBI signal (e/f)	O_SIGNAL CONTROL (c/d)	Information notes	Agent Protocols INAP (k/l)
OriginationAttempt	(o)	-	-	(INAP CS-2) (DP specific)	OriginationAttempt
OriginationAttempt-Authorized	(o)	-	-	(DP specific)	OriginationAttempt-Authorized
OSuspended	(o)	-	-	(INAP CS-2) (DP specific)	OSuspended
Reconnect	(p)	- *)	influence *)	(INAP CS-2) *) ffs	Reconnect
ReleaseCall	(p)	influence	influence		ReleaseCall
ReportUTSI	(o)	influence	influence	(INAP CS-2)	ReportUTSI
RequestCurrentStatus-Report	(p)	-	-		RequestCurrent-StatusReport
RequestEveryStatus-ChangeReport	(p)	-	-		RequestEvery-ChangeReport
RequestFirst-StatusMatchReport	(p)	-	-		RequestFirst-MatchReport
RequestNotification-ChargingEvent	(p)	-*)	*) influence	*)Treatment is national network specific	RequestNotification-ChargingEvent
RequestReport-BCSMEvent	(p)	-influence	-influence	E.g. request for midCall events	RequestReport-BCSMEvent
RequestReportUTSI	(p)	-	-	(INAP CS-2)	RequestReportUTSI
RequestReport-FacilityEvent	(p)	-	-	(INAP CS-2)	RequestReport-FacilityEvent
ResetTimer	(p)	-	-		ResetTimer
RouteSelectFailure	(o)	-	-	(DP specific)	RouteSelectFailure

Table 7-18/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Originating Half Call *(concluded)*

Interface primitives	O_SCF CONTROL (o/p)	IBI signal (e/f)	O_SIGNAL CONTROL (c/d)	Information notes	Agent Protocols INAP (k/l)
SendCharging-Information			influence		
SendFacility-Information	(p)	influence	influence	(INAP CS-2)	SendFacility-Information
SendSTUI	(p)	influence	influence	(INAP CS-2)	SendSTUI
ServiceFiltering-Response	(o)	-	-		ServiceFiltering-Response
SplitLeg	(p)	-	-		SplitLeg

7.2.3.8.2.6 Primitive Signals Mapping Tables, Terminating half call

a) *Call control primitive signals*

Table 7-19/Q.1229 – Primitive Signals, mapping to signalling Agent Protocols, Terminating Half Call

Interface primitives	IBI signal (e/f)	T_SIGNAL CONTROL (h/g)	Information notes	Agent Protocol ISUP (i/j)	Agent Protocol DSS 1 (i/j)
Setup	req.ind (e)	req (g)	End-to-End	IAM	SETUP
Setup	resp.conf (f)	conf (h)	End-to-End *) May be Discarded if backward response sent previously.	ANM, CON *)	CONNECT *)
SubsequentAddress	req.ind (e)	req (g)	Link-by-Link May include both address digits(s) and indication for AddressEnd	SAM	INFORMATION
CallProgress	req.ind (f)	ind (h)	End-to-End *)May be discarded if sent previously	ACM, CPG	ALERTING, PROGRESS

Table 7-19/Q.1229 – Primitive Signals, mapping to signalling Agent Protocols, Terminating Half Call *(concluded)*

Interface primitives	IBI signal (e/f)	T_SIGNAL CONTROL (h/g)	Information notes	Agent Protocol ISUP (i/j)	Agent Protocol DSS 1 (i/j)
Release	req,ind (f)	ind (h)	Link-by-Link B-Party initiated disconnect	REL/RLC	DISCONNECT
Release	req.ind (e)	req (g)	Link-by-Link A-Party (or SSF) initiated disconnect	REL/RLC	DISCONNECT
NetworkSuspend	req.ind (f)	ind (h)	End-to-End CS-2 *) "on-hook"	SUSPEND	- *)
NetworkResume	req.ind (f)	ind (h)	End-to-End CS-2 *) "off-hook"	RESUME	- *)
ServiceFeature	req.ind (f)	ind (h)	"Link-by-Link" B-party initiated Midcall event -- Applies to stimulus and functional terminal protocols	tbd	tbd
ServiceFeature	req.ind (e)	req (g)	A-party initiated Midcall event. -- Applies to stimulus and functional terminal protocols	tbd	tbd
Data	req.ind (f)	ind (h) req (g)	ServiceToUser-Information (Sent to A, received from T_Signal or sent to B)	tbd	tbd
Data	req.ind (e) req.ind (f)	req (g) ind (h)	UserToService-Information (Send by A-Party or B-Party)	tbd	tbd
Failure	ind.req (f) ind.req (e)	ind (h)	*) Call release is initiated by SSF	REL/RLC	DISCONNECT
Reconnect	-	request (g)		ffs	ffs

b) *SCF-SSF primitive signals*

Table 7-20/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Terminating Half Call Table

Interface primitives	T_SCF CONTROL (q/r)	IBI signal (e/f)	T_SIGNAL CONTROL (h/g)	Information notes	Agent Protocols INAP (k/l)
ActivateService-Filtering	(r)	- no influence (except if call is filtered)	- no influence (except if call is filtered)	SSME SSF internal	ActivateService-Filtering
ActivityTest	(r)	-	-	*)SSF-SCF dialogue check	
ApplyCharging	(r)	-	-	-	ApplyCharging
ApplyChargingReport	(q)	-	-	-	ApplyChargingReport
AssistRequest-Instructions	(q)	-	-		AssistRequest-Instructions
AuthorizeTermination	(r)	-influence	influence	(DP specific) CS-2*)- resume terminating basic call process.	AuthorizeTermination
CallGap	(r)	- no influence (except if call is gapped)	- no influence (except if call is gapped)		CallGap
CallInformationReport	(q)	-	-		CallInformation-Report
CallInformation-Request	(r)	-	-		CallInformation-Request
Cancel	(r)	-	-	Cancel "all"	Cancel
CancelStatusReport-Request	(r)	-	-		CancelStatusRequest
Connect	(r)	influence	influence		Connect
ConnectToResource	(r)	-influence	-influence		ConnectToResource
Continue	(r)	-	-		Continue
ContinueWith-Argument	(r)	influence	influence	(INAP CS-2)	ContinueWith-Argument
CreateCSA	(r)	-	-	(INAP CS-2)	CreateCSA

Table 7-20/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Terminating Half Call Table *(continued)*

Interface primitives	T_SCF CONTROL (q/r)	IBI signal (e/f)	T_SIGNAL CONTROL (h/g)	Information notes	Agent Protocols INAP (k/l)
DisconnectForward-Connection	(r)	-	- *)	*) release of e.g. a temporary connection to an IP is not modelled	DisconnectForward-Connection
DisconnectLeg	(r)	influence	influence	(INAP CS-2)	DisconnectLeg
EntityReleased	(q)	-	-	(INAP CS-2)	EntityReleased
EstablishTemporary-Connection	(r)	- *)	- *)	*) set-up of a temporary connection to an IP is not modelled.	EstablishTemporary-Connection
EventNotification-Charging	(q)	-	-		Event-Notification-Charging
EventReportBCSM	(q)	-	-		EventReport-BCSM
EventReport-Facility	(q)	-	-	(INAP CS-2)	EventReportFacility
FacilitySelectedAnd-Available	(q)			(INAP CS-2)	FacilitySelected-AndAvailable
FurnishCharging-Information	(r)	-	-		FurnishCharging-Information
HoldCallInNetwork	(r)	influence			HoldCallIn-Network
InitialDP	(q)	-	-		InitialDP
InitiateCallAttempt	(r)	influence	influence		InitiateCall-Attempt
ManageTriggerData	(r)	-	-	(INAP CS-2)	ManageTriggerData
MergeCallSegments	(r)	-	-	(INAP CS-2)	MergeCallSegments
MoveCallSegments	(r)	-	-	(INAP CS-2)	MoveCall-Segments
MoveLeg	(r)	-	-	(INAP CS-2)	Moveleg
Reconnect	(r)	-	- *) ffs	(INAP CS-2)	Reconnect
ReleaseCall	(r)	influence	influence		ReleaseCall
ReportUTSI	(q)	influence	influence	(INAP CS-2)	ReportUTSI

Table 7-20/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Terminating Half Call Table *(continued)*

Interface primitives	T_SCF CONTROL (q/r)	IBI signal (e/f)	T_SIGNAL CONTROL (h/g)	Information notes	Agent Protocols INAP (k/l)
RequestCurrent-StatusReport	(r)	-	-		RequestCurrent-StatusReport
RequestEveryStatus-ChangeReport	(r)	-	-		RequestEvery-ChangeReport
RequestFirstStatus-MatchReport	(r)	-	-		RequestFirst-MatchReport
RequestNotification-ChargingEvent	(r)	influence *)	-*)	*)Treatment is national network specific	RequestNotification-ChargingEvent
RequestReport-BCSMEvent	(r)	-influence	-influence	E.g. request for midCall events	RequestReport-BCSMEvent
RequestReportUTSI	(r)	-	-	(INAP CS-2)	RequestReportUTSI
RequestReport-FacilityEvent	(r)	-	-	(INAP CS-2)	RequestReport-FacilityEvent
ResetTimer	(r)	-	-		ResetTimer
SelectFacility	(r)	influence	influence	resumes terminating basic call processing to select line	SelectFacility
SendCharging-Information	(r)	influence	-		SendCharging-Information
SendFacility-Information	(r)	influence	influence		
SendSTUI	(r)	influence	influence	(INAP CS-2)	SendSTUI
ServiceFiltering-Response	(q)	-	-		ServiceFiltering-Response
SplitLeg	(r)	-	-	(INAP CS-2)	SplitLeg
TAnswer	(q)	-	-	(DP specific)	TAnswer

Table 7-20/Q.1229 – Primitive Signals, mapping to SCF Agent Protocols, Terminating Half Call Table *(concluded)*

Interface primitives	T_SCF CONTROL (q/r)	IBI signal (e/f)	T_SIGNAL CONTROL (h/g)	Information notes	Agent Protocols INAP (k/l)
TBusy	(q)	-	-	(DP specific)	TBusy
TDisconnect	(q)	-	-	(DP specific)	TDisconnect
TerminationAttempt	(q)	-	-	(INAP CS-2)	TerminationAttempt
TermAttempt-Authorized	(q)	-	-	(DP specific)	TermAttempt-Authorized
TMidCall	(q)	-	-	(DP specific)	TMidCall

Table 7-21/Q.1229 – Mapping of IBI Signalling Events to Signalling Messages

IBI Signalling Semantics	SETUP req. ind (e)	SETUP resp. conf. (f)	CALL-PROGRESS req. ind. (f)	CALL-PROGRESS req.ind (Alerting) (f)	RELEASE req. ind. (e, f)
Request for connection	X				
Connection accepted by user		X			
Call info complete		X	X	X	
Called user being alerted				X	
Connection unavailable					X
Request to terminate call					X

7.2.3.9 INAP addressing: How to do it and why

Each FE instance specified in the Intelligent Network series of Recommendations must have a unique INAP address. This address must contain the address elements (refer to 18.2.2.1.1/Q.1228) which must be presented to the lower layer network services (TC, SCCP, MTP) in order for the INAP message PDUs to be correctly transferred between communicating FEs. For international internetwork operations to succeed, the INAP address of each of the FEs involved in the operation must be known to the networks involved in transferring the operation messages.

For IN CS-1 (1995), these internetworking interfaces are:

- SCF-SDF

For IN CS-2 (1997), these internetworking interfaces are:

- SCF-SDF
- SDF-SDF
- SCF-SCF.

At the INAP level the internetwork relationship can be represented as shown in Figure 7-42.

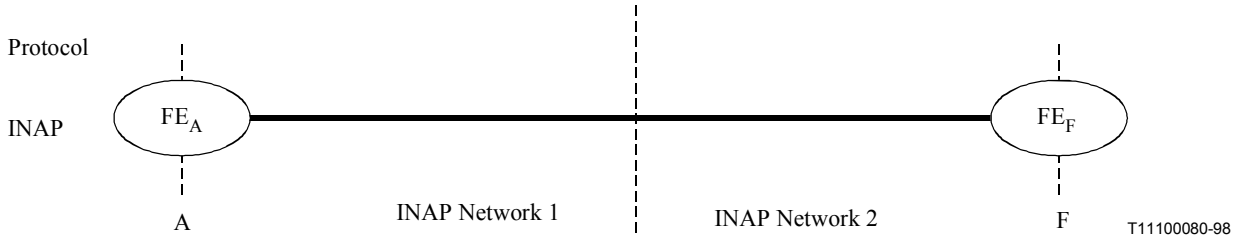


Figure 7-42/Q.1229 – INAP internetwork connection between FEs

Table 7-22 identifies the permissible FE types for each of the relevant IN capability sets:

Table 7-22/Q1229 – Permissible FEs for internetwork operations

Capability Set	FE_A	FE_F
IN CS-1 (1995)	SCF	SDF
IN CS-2 (1997)	SCF	SDF
	SDF	SDF
	SCF	SCF

7.2.3.9.1 Mapping of INAP FEs to SCCP subsystems

INAP uses the addressing services of the lower layer network services [TC, SCCP, MTP] to ensure the correct delivery of INAP PDU messages. Since TC does not manipulate any of the addressing elements, it is in effect the SCCP address parameters which will control how INAP messages are handled for message delivery. Each INAP network node will contain a single SCCP layer which handles the routing of network messages. The SCCP will deliver messages to an identified subsystem (which is addressed by a subsystem number, SSN). This subsystem corresponds to an Application Entity (AE) which contains the TC ASE plus all of the application layer function ASEs to be performed in the node.

Recommendation Q.1225 contains the agreed set of mappings of FEs to physical network nodes for IN CS-2 (1997). It is possible for IN network nodes to contain one or more FEs. The mapping of these FEs into specific AE types which can be addressed as subsystems by the SCCP has not been standardized and may in fact be implementation dependent. Figure 7-43 shows possible mappings of FEs to AEs and the effect this has on the SCCP addressing. The configurations shown are not considered to be exhaustive.

It should be noted that a FE (as used in IN terminology) is equivalent to an AE (as defined in Recommendation Q.1400; in Figure 7-43 the model ii) shows the example where AE type (FE type) can have two different applications of the same type in one node.

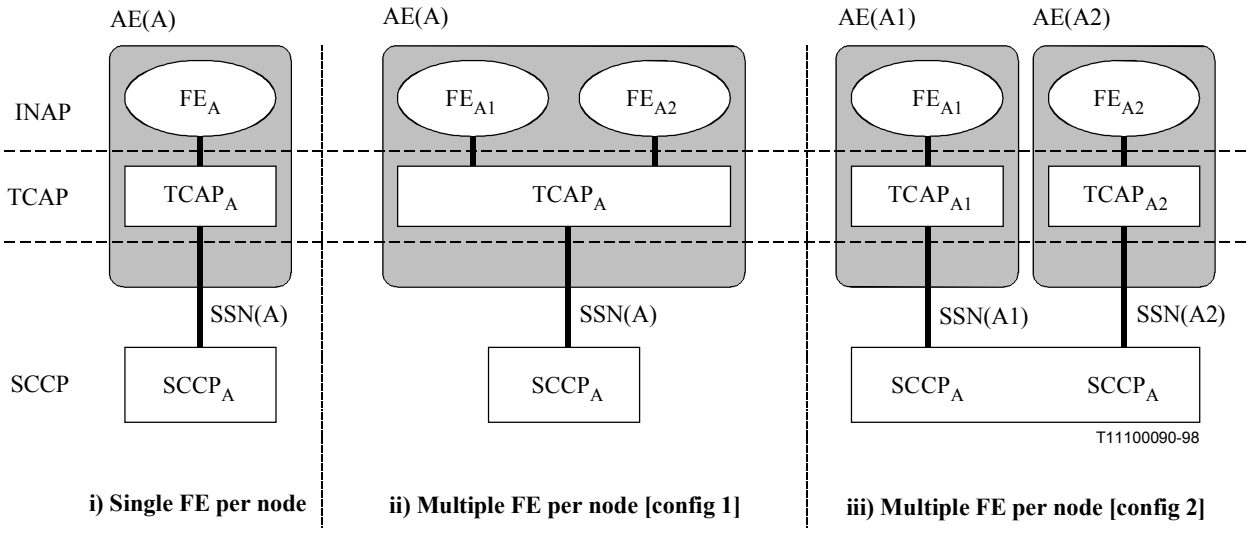


Figure 7-43/Q.1229 – Possible mappings of FEs to AEs

7.2.3.9.2 Interconnection of INAP nodes for internetworking

The INAP addresses are constrained to fit into the SCCP Called Party Address structure defined in Recommendation Q.713. However, to fully realize the consequences of selecting a specific form of this address for INAP internetworking use, we need to examine the full physical level interaction between INAP and the lower layer routing functions.

An example internetwork configuration showing this is given in Figure 7-44.

The configuration in Figure 7-44 was chosen because it is the simplest network configuration which illustrates the minimum requirements for international internetwork operations. While other configurations are possible they would not simplify the requirements imposed on the addressing needs for INAP operations.

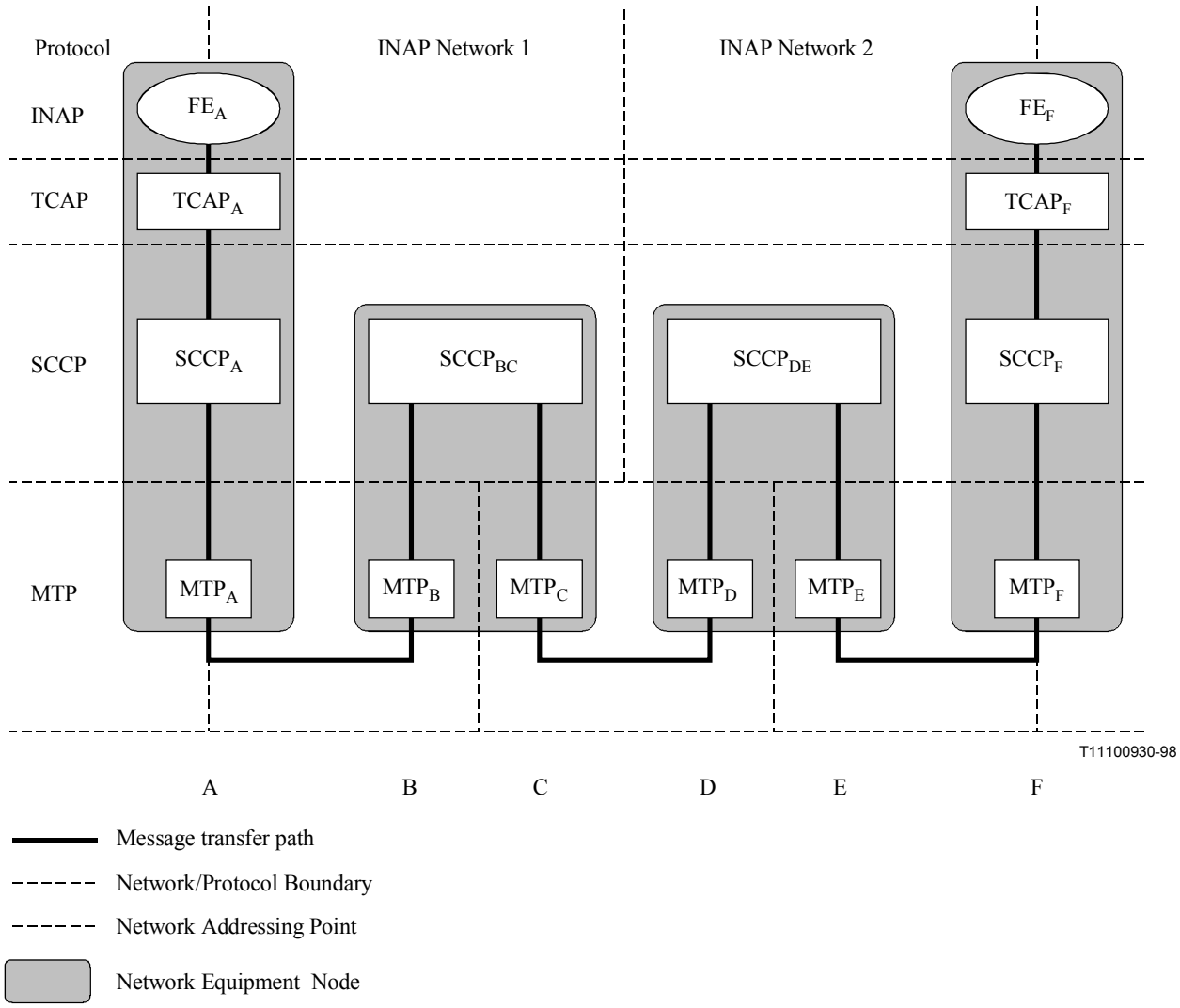


Figure 7-44/Q.1229 – Model for internetwork connection between FEs

7.2.3.9.3 Address Mapping during message transfer

To illustrate the mapping of addressing parameters between the different protocol networks, the establishment of a TC dialogue between FE_A and FE_F will be examined.

The format used for the messages uses the parameters defined in the appropriate Recommendations. The values of the parameters which affect the addressing requirements are in **bold** typeface. Table 7-23 defines the abbreviations used for the parameters in the messages.

Table 7-23/Q.1229 – Abbreviations for element values for message events

Parameter	Description	Protocol
QoS	Quality of Service (as defined in 3.1.2.1/Q.771)	TC
ac(X)	Application Context name value X (ASN.1 object id)	TC
dtid(X)	Destination Transaction ID value X (as defined in 3.1/Q.773)	TC
otid(X)	Originating Transaction ID value X (as defined in 3.1/Q.773)	TC
BEGIN{X}	TC Message BEGIN (ASN.1 defined in 3.1 and 3.2/Q.773)	TC
CONTINUE{X}	TC Message CONTINUE (ASN.1 defined in 3.1 and 3.2/Q.773)	TC
dialogue{X}	TC message dialogue portion (ASN.1 defined in 3.2/Q.773)	TC
components(X)	TC message components portion (defined in 4.2.2/Q.773)	TC
tid(X)	Transaction ID (Dialogue ID) value X (Q.771)	TC
UI	User Information (defined in 4.2.3/Q.772)	TC
adrX(Z)	SCCP address value X (defined in 3.4/Q.713) X = S for primitive (SDU) where formatting is not specified X = P for protocol (PDU) where formatting is specified	SCCP
class()	Protocol Class (as defined in 3.6/Q.713)	SCCP
gt(X)	Global Title value X (as defined in 3.4.2.3/Q.713)	SCCP
hc()	Hop Counter (as defined in 3.18/Q.713)	SCCP
pc(X)	Point Code value X (as defined in 3.4.2.1/Q.713)	SCCP
Ret	Return option (as defined in 6.2.2.2.3/Q.711)	SCCP
rgt	route on GT	SCCP
rpc	route on SSN	SCCP
X(L)(X)UDT{X}	SCCP Message X(L)(X)UDT value X (format defined in 4.10/Q.713 for UDT, 4.18/Q.713 for XUDT and 4.20/Q.713 for LUDT)	SCCP
seg(X)	Segmentation value X (defined in 3.17/Q.713)	SCCP
Seq	Sequencing (defined in 6.2.2.2.2/Q.711)	SCCP
ssn(X)	SSN value X (as defined in 3.4.2.2/Q.713	SCCP
dpc(X)	Destination Point Code value X (as defined in 2.2/Q.704)	MTP
opc(X)	Originating Point Code value X (as defined in 2.2/Q.704)	MTP
sio(N,X)	Service Information Octet with Network Indicator value N and Service Indicator value X (as defined in 14.2/Q.704)	MTP
sls(X)	Signalling Link Selection value X (as defined in 2.2/Q.704)	MTP

In the subsequent clauses, a value of (?) indicates that the value is set by mechanisms contained within the specific protocol. In addition, values denoted by {...} indicate that while the value may be affected by INAP it contains no information which is relevant to the issue of INAP addressing.

7.2.3.9.3.1 Address format for INAP

INAP is required to provide sufficient information to ensure that the SCCP calling and called party addresses are constructed correctly.

For further information, refer to Recommendations Q.713 and Q.714.

7.2.3.9.3.2 Addressing from INAP to TC

INAP uses the TC-User services of TC to transfer messages between INAP FEs.

The TC-User services only require addressing information to be provided during the establishment of a TC dialogue between the two end TC-Users. Addresses are carried in the TC-BEGIN primitive (originating address and destination address*)* and in the first TC-CONTINUE primitive (originating address). Because TC does not specify any address manipulation, the form of these addresses must be compatible with the SCCP addresses used in the layer below TC. The originating address parameter values in both the TC-BEGIN and the first TC-CONTINUE must be unambiguous in that they must uniquely identify the respective TC nodes.

In addition to the address information, the dialogue establishment uses an application context name (AC) which is used to identify the set (and direction) of operations which can be transported by the dialogue.

The IN Physical plane (Recommendations Q.1215 and Q.1225) allows for the co-location of a number of INAP FEs within a single network equipment node. It is the responsibility of the "glue" between TC and the specific FE (as contained within the SACF/MACF functions) to associate incoming TC dialogues (indicated by a TC-BEGIN.ind primitive) with the appropriate FE.

From the parameters provided by the TC-User service, the SACF/MACF can perform this function by examining the destination address elements and/or the AC. For example:

1) The destination FE can be determined from the destination address, one for each FE at the node.

   This implies that each equipment node will potentially require multiple INAP addresses.

2) The destination FE can be determined from the value of the AC (after the address has been used to deliver the message to the SCCP subsystem) e.g. for message routing from an SCF to SDF, then the SDF node examining the AC (SCF-SDF) knows the message is for the destination SDF FE.

   This implies that each equipment node will require only a single address and that the AC can uniquely identify the destination FE. Note that this method will not work if more than one FE of a particular type (e.g. two or more SCFs) are co-located in the same equipment node.

For the example in Figure 7-44, the TC level information exchange is detailed in Table 7-24. The information flows for this exchange are shown in Figure 7-45.

Table 7-24/Q.1229 – Message definitions for TC level message exchange

Step	From	To	Message format
1	FE_A	TC_A	TC-BEGIN.req (QoS, adrS(**inap[FE-F1]**), ac(**AF**), adrS(**inap[FE-A]**), tid(**A1**), UI);
2	TC_A	TC_F	BEGIN{ otid(**A1**), dialogue { ac(**AF**), UI, ... }, component {...} };
3	TC_F	FE_F	TC-BEGIN.ind (QoS, adrS(**inap[FE-F4]**), ac(**AF**), adrS(**inap[FE-A]**), tid(**F1**), UI);
4	FE_F	TC_F	TC-CONTINUE.req (QoS, adrS(**inap[FE-F]**), ac(**AF**), tid(**F1**), UI);
5	TC_F	TC_A	CONTINUE{ otid(**F1**), dtid(**A1**), dialogue { ac(**AF**), UI, ... }, component {...} };
6	TC_A	FE_A	TC-CONTINUE.ind (QoS, , tid(**A1**), UI);
NOTE – In steps 2 and 5 the TC data is transferred using the services of the SCCP.			

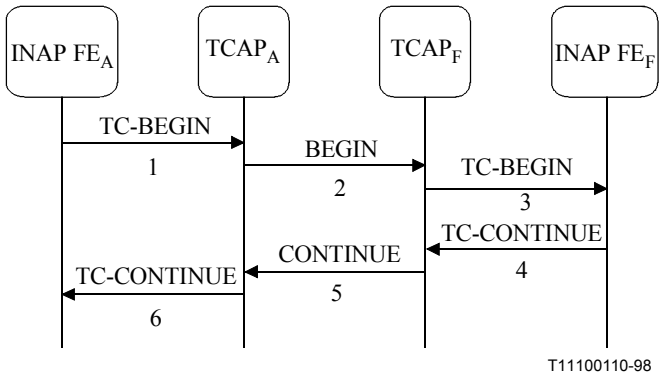


Figure 7-45/Q.1229 – Information flows for TC level message exchange

The TC-BEGIN primitive requires the following parameters relating to addressing:

originating address	adrS(**inap[FE-A**])
destination address	adrS(**inap[FE-F1]**)
application context name	ac(**AF**)

The initial TC-CONTINUE primitive requires the following parameter relating to addressing:

originating address	adrS(**inap[FE-F**])

These address parameters have the following constraints on their values:

- **inap[FE-F1]**, **inap[FE-A]** and **inap[FE-F]** must conform[2] to the SCCP address formats (see 7.2.3.9.3.1).
- **inap[FE-F1]** is an address which must map to a set of INAP FEs that provide identical INAP functionality. The selection of which FE actually performs the required function is left to the SCCP layer to determine.
- **inap[FE-A]** and **inap[FE-F]** must unambiguously identify the TC nodes in the international IN address space. If the AC is not used to determine the destination FE, then the addresses must unambiguously identify the individual INAP FEs.
- **AF** must take the value of one of the INAP ACs defined for the specific internetworking interface being used (SCF-SDF, SDF-SDF, SCF-SCF).

7.2.3.9.3.3 Addressing from TC to SCCP

TC uses the N-UNITDATA service of the SCCP-User services of SCCP to transfer messages between TC nodes. Addresses are carried by this service in the *called party address* and the *calling party address* parameters. Once in the SCCP network, messages are routed between SCCP nodes until the destination SCCP node is reached. This routing is performed in one of two ways:

- *route on GT*, the global title translation has to derive just a point code at all but the last translation (at the last translation additionally a SSN is identified, either from GT translation or by using the SSN possibly included as a separate address element in the called party address). Exceptionally a translation may also produce a new destination called party address, but in this case it should be noted that the application's ability to act on messages that are returned on error by the SCCP may be affected.

---

[2] While primitives do not specify exact formats, they must contain the information elements required to populate the SCCP formatted addresses.

- *route on SSN*, which results in the message being transferred to the specified subsystem at the end SCCP node. Once this form of routing is selected, for example in the output from a Global Title Translation (GTT) function, then no further address translations are permitted.

As there is the possibility of duplication of point codes between networks, messages that cross network boundaries are required to be routed according to GT, at least up to the last translation node. When messages cross the international boundary, the allowed formats of the address parameters are restricted according to the rules detailed in Annex B/Q.713. See also Recommendation Q.715 (SCCP user guide). In particular the use of *route on SSN* for SCCP message routing in the originating network is only possible if the originating node is in the international network and if:

a) the final destination SCCP node is visible in the international SCCP network;

b) there is a standardized non-zero SSN value for the INAP service.

It is possible to satisfy condition a) by co-locating the INAP/TC node with the international SCCP node. However, given the limited number of international SCCP addresses available to individual network providers[3], it is more likely that network configurations similar to that shown in Figure 7-44 will be used.

It is therefore mandatory, even where condition a) is met, to use the route on GT mechanism.

For the example in Figure 7-44 the SCCP level message exchange is detailed in Table 7-25. The information flows for this message exchange are shown in Figure 7-46. The XUDT message was used for this example and, although other SCCP message formats could be used (e.g. UDT, LUDT). It should be noted that currently only UDT messages are widely supported.

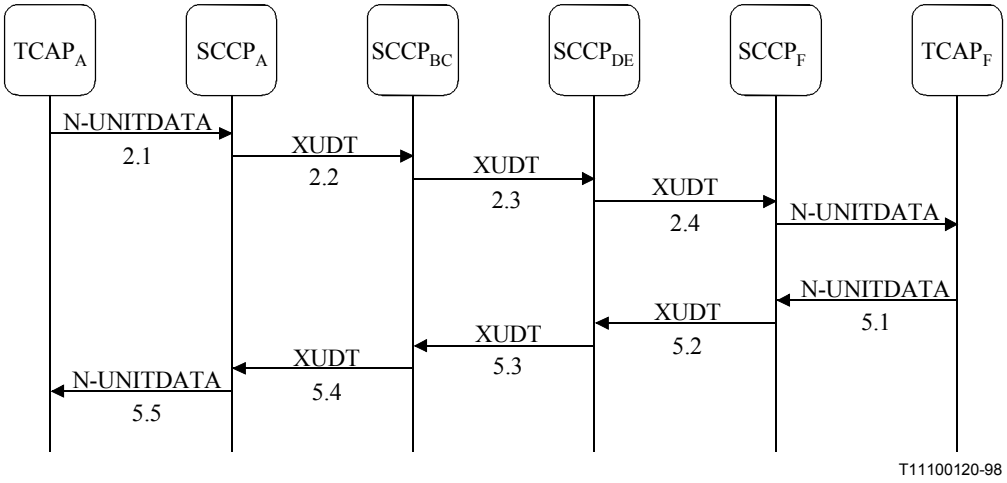


Figure 7-46/Q.1229 – Information flows for SCCP level message exchange

[3] As defined in Recommendation Q.708.

Table 7-25/Q.1229 – Message definitions for SCCP level message exchange

Step	From	To	Message format
2.1	TC_A	$SCCP_A$	N-UNITDATA.req (adrS(**inap[FE-F1]**), adrS(**inap[FE-A]**), Seq, Ret, Imp, BEGIN{...});
		GTT_A	**inap[FE-F1]** translates to { MTP-SAPi(MTP Network 1), dpc(**B**), adrP(**inap[FE-F2]**) }
2.2	$SCCP_A$	$SCCP_{BC}$	XUDT {adrP(**inap[FE-F2]**), adrP(**inap[FE-A]**), BEGIN{...}, …}
		GTT_{BC}	**inap[FE-F2]** translates to { MTP-SAPi(MTP Int Nw), dpc(**D**), adrP(**inap[FE-F3]**) }
2.3	$SCCP_{BC}$	$SCCP_{DE}$	XUDT {adrP(**inap[FE-F3]**), adrP(**inap[FE-A]**), BEGIN{...}, …}
		GTT_{DE}	**inap[FE-F3]** translates to { MTP-SAPi(MTP Network 2), dpc(**F**), adrP(**inap[FE-F4]**) }
2.4	$SCCP_{DE}$	$SCCP_F$	XUDT {adrP(**inap[FE-F4]**), adrP(**inap[FE-A]**), BEGIN{...}, …}
2.5	$SCCP_F$	TC_F	N-UNITDATA.ind (adrS(**inap[FE-F4]**), adrS(**inap[FE-A]**), Seq, Ret, Imp, BEGIN {...});
5.1	TC_F	$SCCP_F$	N-UNITDATA.req (adrS(**inap[FE-A]**), adrS(**inap[FE-F]**), Seq, Ret, Imp, CONTINUE{...});
		GTT_F	**inap[FE-A]** translates to { MTP-SAPi(MTP Network 2), dpc(**E**), adrP(**inap[FE-A2]**) }
5.2	$SCCP_F$	$SCCP_{DE}$	XUDT {adrP(**inap[FE-A2]**), adrP(**inap[FE-F]**), CONTINUE{...}, …}
		GTT_{DE}	**inap[FE-A2]** translates to { MTP-SAPi(MTP Int Nw), dpc(**C**), adrP(**inap[FE-A3]**) }
5.3	$SCCP_{DE}$	$SCCP_{BC}$	XUDT {adrP(**inap[FE-A3]**), adrP(**inap[FE-F]**), CONTINUE{...}, …}
		GTT_{BC}	**inap[FE-A3]** translates to { MTP-SAPi(MTP Network 1), dpc(**A**), adrP(**inap[FE-A4]**) }
5.4	$SCCP_{BC}$	$SCCP_A$	XUDT {adrP(**inap[FE-A4]**), adrP(**inap[FE-F]**), CONTINUE{...}, …}
5.5	$SCCP_A$	TC_A	N-UNITDATA.ind (adrS(**inap[FE-A4]**), adrS(**inap[FE-F]**), Seq, Ret, Imp, CONTINUE{...});
NOTE 1 – In steps 2.2, 2.3, 2.4, 5.2, 5.3 and 5.4, the SCCP data is transferred using the services of the MTP.			
NOTE 2 – Seq, Ret and Imp shown in steps 2.5 and 5.5 are optional.			

The N-UNITDATA primitive carrying the TC-BEGIN requires the following parameters relating to addressing:

calling address adrS(**inap[FE-A**])

called address adrS(**inap[FE-F1]**).

Global Title Translations are performed within $Node_A$, $Node_{BC}$ and $Node_{DE}$ and possibly in Node F. Each of these translations may exceptionally produce a new called party address for use in the SCCP message transfer to the next node. These addresses (changed by the translation or not) are denoted as **inap[FE-F2]**, **inap[FE-F3]** and **inap[FE-F4]** respectively.

The N-UNITDATA primitive carrying the first TC-CONTINUE requires the following parameters relating to addressing:

calling address adrS(**inap[FE-F**])

called address adrS(**inap[FE-A]**)

For more details about the information contained in the called and calling addresses, refer to Appendix II.

It should be noted that Table 7-25 is in need of modification. It would be useful to clarify that address elements indicated as contained in a parameter of an "adrS" are not literal, but instead represent information that maps to the equivalent address element in a parameter of an "adrP".

7.2.3.9.3.4 Addressing from SCCP to MTP

SCCP uses the MTP-TRANSFER service of the MTP-User services of MTP to transfer messages between SCCP nodes. Addresses are carried by this service in the origin point code and the destination point code parameters.

For the example in Figure 7-44 the MTP level message exchanges are detailed in Table 7-26. The information flows for this message exchange are shown in Figure 7-47.

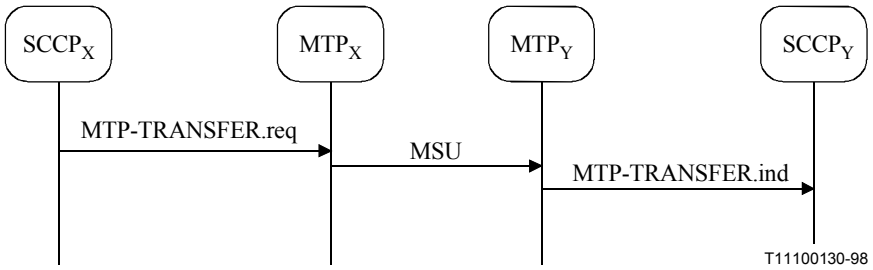


Figure 7-47/Q.1229 – Information flows for example MTP level message exchange

Table 7-26/Q.1229 – Message definitions for SCCP level message exchange

Step	From	To	Message format
2.2.1	$SCCP_A$	MTP_A	MTP-TRANSFER.req (opc(A), dpc(B), sls(?), sio(?,3), XUDT{...});
2.2.2	MTP_A	MTP_B	MSU { sio(?,3), dpc(B), opc(A), sls(?), XUDT{...} };
2.2.3	MTP_B	$SCCP_{BC}$	MTP-TRANSFER.ind (opc(A), dpc(B), sls(?), sio(?,3), XUDT{...});
2.3.1	$SCCP_{BC}$	MTP_C	MTP-TRANSFER.req (opc(C), dpc(D), sls(?), sio(?,3), XUDT{...});
2.3.2	MTP_C	MTP_D	MSU { sio(?,3), dpc(D), opc(C), sls(?), XUDT{...} };
2.3.3	MTP_D	$SCCP_{DE}$	MTP-TRANSFER.ind (opc(C), dpc(D), sls(?), sio(?,3), XUDT{...});
2.4.1	$SCCP_{DE}$	MTP_E	MTP-TRANSFER.req (opc(E), dpc(F), sls(?), sio(?,3), XUDT{...});
2.4.2	MTP_E	MTP_F	MSU { sio(?,3), dpc(F), opc(D), sls(?), XUDT{...} };
2.4.3	MTP_F	$SCCP_F$	MTP-TRANSFER.ind (opc(E), dpc(F), sls(?), sio(?,3), XUDT{...});
5.2.1	$SCCP_F$	MTP_F	MTP-TRANSFER.req (opc(F), dpc(E), sls(?), sio(?,3), XUDT{...});
5.2.2	MTP_F	MTP_E	MSU { sio(?,3), dpc(E), opc(F), sls(?), XUDT{...} };
5.2.3	MTP_E	$SCCP_{DE}$	MTP-TRANSFER.ind (opc(F), dpc(E), sls(?), sio(?,3), XUDT{...});
5.3.1	$SCCP_{DE}$	MTP_D	MTP-TRANSFER.req (opc(D), dpc(C), sls(?), sio(?,3), XUDT{...});
5.3.2	MTP_D	MTP_C	MSU { sio(?,3), dpc(C), opc(D), sls(?), XUDT{...} };
5.3.3	MTP_C	$SCCP_{BC}$	MTP-TRANSFER.ind (opc(D), dpc(C), sls(?), sio(?,3), XUDT{...});
5.4.1	$SCCP_{BC}$	MTP_B	MTP-TRANSFER.req (opc(B), dpc(A), sls(?), sio(?,3), XUDT{...});
5.4.2	MTP_B	MTP_A	MSU { sio(?,3), dpc(A), opc(B), sls(?), XUDT{...} };
5.4.3	MTP_A	$SCCP_A$	MTP-TRANSFER.ind (opc(B), dpc(A), sls(?), sio(?,3), XUDT{...});

The point code values required have the following constraints:

- pc(**A**), pc(**B**) are standard point codes (3.4.2.1/Q.713) defined by the provider of MTP Network 1.
- pc(**C**), pc(**D**) are international signalling point codes as defined in Recommendation Q.708.
- pc(**E**), pc(**F**) are standard point codes (3.4.2.1/Q.713) defined by the provider of MTP Network 2.

7.2.3.9.4 Summary of Protocol Requirements for INAP Addressing

The following requirements can be identified for the different protocols used to carry INAP internetwork messages:

7.2.3.9.4.1 Requirements on INAP

1) If Application Contexts (ACs) are to be used for FE differentiation, then the values assigned to the INAP AC must uniquely identify the destination FE.

   This requirement applies to all interfaces, not just those used for internetworking.

2) If AC values are not used for FE differentiation, then an agreed format for the Global Title must be defined at the INAP level for international addressing.

   These may conform to the formats specified in B.4.3/Q.713 or B.4.4/Q.713. If an alternative numbering scheme is required, then this must be forwarded to ITU-T for inclusion in a future revision of Annex B/Q.713.

7.2.3.9.4.2 Requirements on TC

1) If Application Contexts (ACs) are to be used for FE differentiation within a physical node, then the version of TC used must support the dialogue portion of TC (i.e. 1993 revision of TC).

   This requirement applies to all interfaces, not just those used for internetworking.

7.2.3.9.4.3 Requirements on SCCP

1) The network provider must ensure that any change of GT value during translation preserves any INAP specific information contained in the initial GT value.

   This requirement applies to all interfaces, not just those used for internetworking.

2) If route on SSN is to be supported from the originating node, then a non-zero internationally standardized SSN is required for international internetworking (currently not agreed and not standardized).

   NOTE 1 – For this the originating node is also required to be in the international network.

   In the absence of a standardized non-zero SSN for INAP services, the use of route on GT is mandatory from the origin node to the network containing the destination node where network boundaries are crossed.

3) The version of SCCP used to support INAP operations must be at least SCCP 1993 if segmentation/reassembly of messages is required to be performed by the SCCP. The need for SCCP 1996 (for improved congestion control mechanisms) is still to be determined.

   NOTE 2 – At present SCCP 1993 is not widely supported. As a consequence, INAP operations that cross network boundaries should not assume SCCP 1993 capability.

The SCCP requirements are those required to support INAP operations. Currently there is nothing being done on determining the requirements of INAP Management.

SCCP Management consists of a number of services (N-COORD, N-STATE, N-PCSTATE) all of which will be presented to the INAP message interface. These services allow the user to notify the SCCP of the availability of subsystems within the SCCP.

Since the use of these services requires the specification of a non-zero subsystem number, they currently cannot be used for INAP management operations in international internetworking situations.

It should be noted that current SCCP 1996 is not particularly clear on the management procedures involving multiple SCCP networks.

7.2.3.9.4.4 Requirements on MTP

There are no INAP specific requirements of the parameters of the MTP. Existing MTP requirements for international addressing are sufficient.

7.2.3.9.5 Implications for network providers

In this subclause some of the implications of the requirements identified in 7.2.3.9.3.1 for network providers will be examined.

7.2.3.9.5.1 Effect of Addressing on message payload

The carriage of INAP messages between nodes is subject to the message size limitations imposed by the lower layers used. Currently, INAP uses a protocol stack as shown in Figure 7-48.

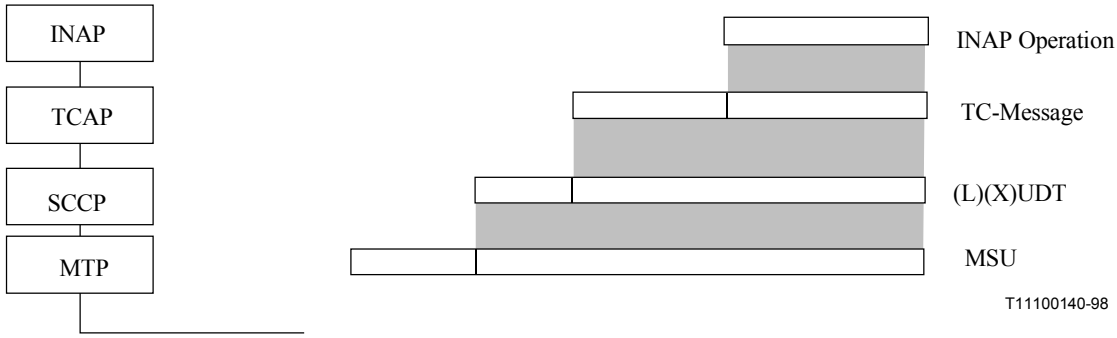


Figure 7-48/Q.1229 – INAP Protocol Stack

The available versions of the lower protocols are shown below:

TCAP: TCAP 1988, TCAP 1993

SCCP: SCCP 1988, SCCP 1993, SCCP 1996

MTP: MTP 1988, MTP 1993

NOTE 1 – The SCCP UDT message is available in all versions of the Recommendations.

The SCCP XUDT message is available in the SCCP 1993 and SCCP 1996 versions of the Recommendations.

The SCCP LUDT message is only available in the SCCP 1996 version of the Recommendations.

Table 7-27 summarizes the maximum message payload sizes for user defined portions of the various SCCP protocol used:

Table 7-27/Q.1229 – Maximum size (octets) for Addresses + User Data in SCCP messages

SCCP message	Blue Book SCCP	SCCP 1993	SCCP 1996
UDT	260	260	260
XUDT (single)	–	2587	254
XUDT (multiple) –[a)]	–	251	248
LUDT	–	–	3968
[a)] A maximum of 16 XUDT messages can be used to carry User Data.			

In each case, the maximum message payload contains:

- the Addresses portion which consists of the CallingPartyAddress and CalledPartyAddress encoded according to the SCCP CalledPartyAddress formats;
- the User Data portion which consists of the encoded PDU containing the TC Message. INAP operations will be encoded into the TC Message.

NOTE 2 – In the case where the User Data portion is carried in a number of XUDT messages, then all of the XUDT messages used will have the same address portion value.

The length of a XUDT message[4] is a combination of fixed overhead, addresses and data. The net effect on message payloads is that:

- Addresses + part(Data) ≤ 248 octets for segmented messages
  (data split over up to 16 XUDT messages);
- Addresses + Data ≤ 254 octets for non-segmented messages
  (data is carried in a single XUDT message).

For international SCCP addresses, two possible alternatives are currently prescribed, either international E.164 or Generic number with Q.708 Z-UUU-V prefix.

The maximum size of an E.164 international address is 15 digits. Using BCD encoding this puts the maximum length of an E.164 based address at 15 octets. Therefore using E.164 addresses message payload constraints are:

- max length Data ≤ 224 octets for single XUDT messages;
- max length Data ≤ 3456 octets for multiple XUDT messages using segmentation.

The maximum size of the Generic number depends on the contents of the national portion. The fixed part of the address for this form requires 9 octets. In this case there must be agreement between the network providers as to the maximum address length in order to ensure that data is not lost due to changes in the address size (refer to 7.2.3.9.5.2).

7.2.3.9.5.2 Effect of Addressing on Global Title Translation tables

Currently SCCP GTT routing must be used to direct international internetworking traffic.

At each translation point the SCCP may exceptionally generate a new called party address for the message being transferred. It is therefore possible for the size of the address to change as the SCCP message is passed from node to node. If at one of these translation points the new size of the address is such that the SCCP message can no longer fit into the MTP payload, then the delivery of the message will fail. Network operators should ensure that sufficient space in the SCCP message is provided by the initial SCCP node to carry the maximum sized address.

Additionally, in the absence of any requirement about the use of the TC level AC to determine the destination INAP FE, it would be prudent for the network operator to assume that the GT portion of any SCCP contains information required by the INAP. This would require that the GT portion of each address should be preserved during the translation process including the last translation node, which will require careful population and management of the GTT tables in each SCCP node. Particular care must be taken in the case where the SCCP is used to select one of a set of nodes used to provide the same service. In this case the network provider must ensure that the change of GT value preserves the INAP specific information contained in the initial GT value.

While a SCCP node performing GTT on a message could modify only the called party address of the message, the first international gateway SCCP node may also modify the calling party address in order to make it conform to international requirements. In this case the network provider must ensure that any INAP related information is retained during the modification and that the resulting address is still unambiguous (i.e. it only identifies a single entity in the network).

7.2.3.9.6 INAP Address Format for international interworking

The two alternatives for INAP addresses for international internetworking are:

1) Global Title type 4 containing E.164 address with Translation Type = 0;

---

[4] As used in the example in 7.2.3.9.3.

2) Global Title type 4 containing Generic Number with Q.708 prefix and Translation Type = 2.

Because of the potential need to address multiple instances of a particular type of INAP FE (e.g. multiple special purpose SDFs) within a specific network node, it is necessary for the GT portion of the SCCP address to identify uniquely the INAP FE being addressed.

For a GT containing an E.164 address this effectively mandates that each instance of an INAP FE has its own nationally significant E.164 address.

The alternative to assigning an E.164 address to each INAP FE is to use a GT based on the Generic number plan version of the international SCCP address. Applying INAP requirements to such a generic number should result in a global title of the form:

ZUUUV NNNNNN FF

where ZUUUV is the Q.708 portion, NNNNN is the national significant number which identifies the network node, FF identifies the INAP AE instance within the node.

If this format is chosen, then it may be necessary to have a new international Translation Type and/or international SSN defined for INAP in order for INAP specific addresses (Generic Number + AP FE Suffix) to be distinguished from non-INAP specific addresses based on the generic numbering plan which could employ identical address digits.

Figure 7-49 shows an example coding of such an address using SCCP formats. The INAP Id digits are used to address the specific INAP FE functionality within the SCCP subsystem to which the message was delivered.

Note that in the example below, the information up to and including octet 8 is the international standardized part of the address, and octets 9-N form the non-standardized national part of the address.

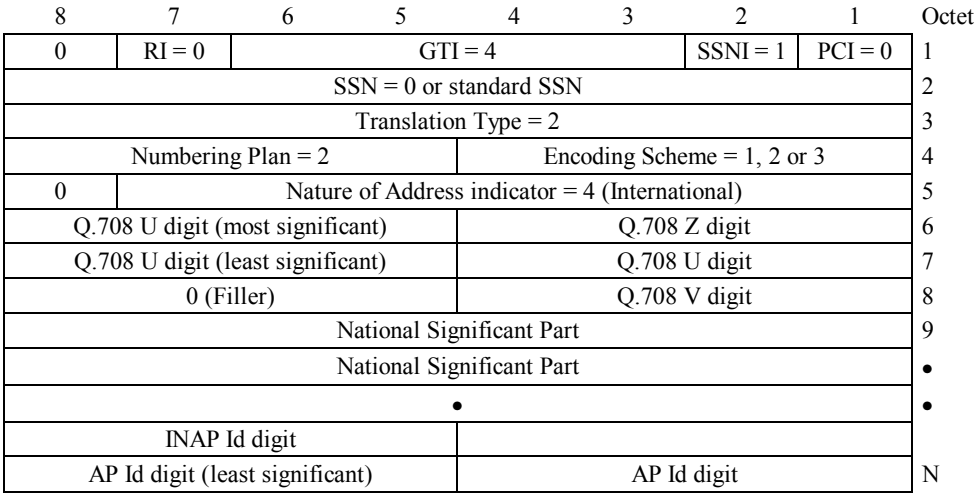


Figure 7-49/Q.1229 – Address format for INAP FE for internetworking

7.2.3.10 IN CS-2 flow control mechanism

This subclause summarizes the flow control mechanisms specified in the IN CS-2 Recommendations. These mechanisms may not be sufficient for some cases of IN network congestion. More advanced and effective INAP flow control mechanisms will be provided in IN CS-3 or latter IN capability sets.

7.2.3.10.1 Call Gap operation (SCF-SSF)

The Call Gap operation allows the SCF to request the SSF to reduce traffic on the basis of service invoked, called party, calling party or a combination of these. The operation specifies a duration for which the control is to be applied and an interval between calls.

The operation of the Call Gap traffic control mechanism is shown in Figure 7-50.

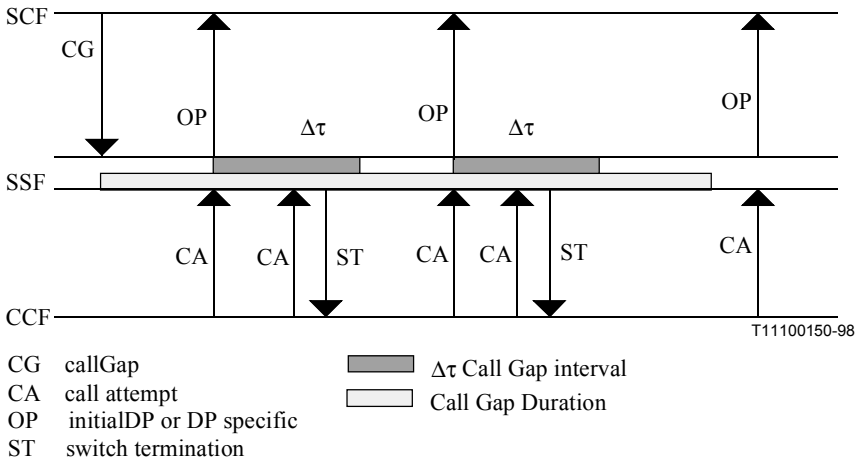


Figure 7-50/Q.1229 – Call Gap flow control

7.2.3.10.2 Service filtering (SCF-SSF)

The service filtering concept is similar to the Call gapping mechanism with several key differences:

- The use of Service filtering is intended to be a normal part of call processing for a service in that it allows switch-based handling suitable for mass calling and televoting class of services.
- The filtering interval definition can be done on number of calls between specific start and stop times. Call Gap starts immediately and can only be done on a time duration and interval.
- Statistical information relating to the number of calls filtered is collected and returned to the SCF at the end of each filtering interval.

The operation of the service filtering traffic control mechanism is shown in Figure 7-51.

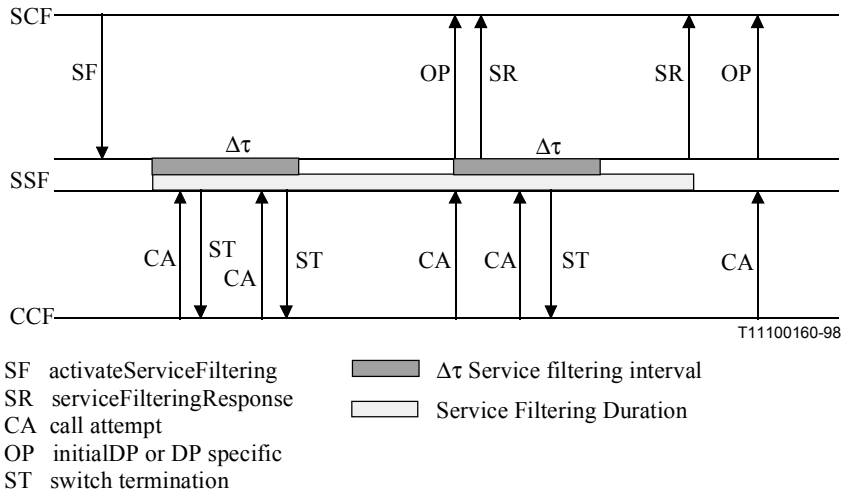


Figure 7-51/Q.1229 – Service Filtering flow control

If service filtering and call gapping are active at the same time for the same criteria, then any call which progress after service filtering will be subject to the call gapping criteria.

7.2.3.11 DP Generic approach and DP Specific approach

The IN CS-2 INAP specifies two options for call modelling associated APDUs. These two options are labelled "DP Generic approach" and "DP Specific approach" and are characterized as follows:

- DP Generic approach:
  - one generic APDU (InitialDP), common to all DPs, initiating the service request. A parameter indicates the DP the service request originated from.
  - generic APDUs per arming mechanism, i.e. Static (TDP, InitialDP) or Dynamic (EDP, EventReportBCSM).
  - the consequence of these two items is as new DPs are defined, new parameters may need to be specified for existing APDUs (InitialDP, EventReportBCSM).

    The receiving entity determines the DP associated with the APDU based on the contents of the received parameter BCSMEventType.

For the SSF-SCF interface, the following DP generic APDUs are supported:

- InitialDP (TDP) and EventReportBCSM (EDP).

- DP Specific approach
  - one unique APDU per DP initiating the service request. The APDU indicates the DP the service request originated from.
  - one APDU independent of the DP arming mechanism, i.e. Static (TDP) or Dynamic (EDP).
  - the consequence of these two items is as new DPs are defined, new APDUs need to be specified.

The receiving entity determines the DP based on received APDU – and DP type (EDP/TDP) based on ServiceAddressInformation as part of dpSpecificCommonParameters.

For the SSF-SCF interface the following DP specific APDUs are supported:

– AnalysedInformation, AnalyseInformation, AuthorizeTermination, CollectedInformation, FacilitySelectedAndAvailable, OAbandon, OAnswer, OCalledPartyBusy, ODisconnect, OMidCall, ONoAnswer, OriginationAttempt, OriginationAttemptAuthorized, OSuspended, RouteSelectFailure, SelectFacility, SelectRoute, TAnswer, TBusy, TDisconnect, TMidCall, TNoAnswer, TerminationAttempt, TermAttemptAuthorized, TSuspended.

For the SCF-SSF interface the following call processing APDUs are used by both the DP Generic and the DP Specific approach.:

– CollectInformation, Connect, Continue, ContinueWithArgument and RequestReportBCSMEvent.

The two options "DP Generic approach" and "DP Specific approach" are mutually exclusive, i.e. either one or the other is supported in an application, but not both.

7.2.3.12 User interaction and CPH processing

IN CS-2 supports two types of user interaction within a Call Segment:

a) UI with SRF resource connected to the Connection Point – allows bidirectional communication path.

b) UI with SRF resource connected to leg – one-way communication path toward user (tone/announcement). This is supported with PlayAnnouncement operation.

CS-2 supports no user interaction during CPH processing, but allows buffering of CPH operations during UI to be executed when e.g. announcement is completed and SRF resource disconnected UI in monitoring (call processing) state.

With IN CS-1, SRF connections can only be made while call processing is suspended at a DP. With IN CS-2, SRF connections can also be made while call processing is not suspended in order to send tone or announcement or display information.

In CS-2 when UI is addressed to a leg, then only tones and announcement and display information sending is to apply to the addressed party, while maintaining the speech connection between that leg and any other leg connected to the same Call Segment.

In CS-2 when UI is addressed toward the Connection Point in the CS, then only tones and announcement sending and display information is to apply to all parties (i.e. SRF connected to the Connection Point) in the Call Segment, while maintaining the speech connection between that leg and any other leg connected to the same Call Segment.

NOTE – If an announcement is sent on one leg (toward one party) the other parties in the call may in "real life" also hear the announcement (but suppressed) due to the reflection caused by the telephone handset.

For User Interaction during call processing, none of the legs connected to the CP may be in setup, i.e. all BCSM instances shall be in the O/T-Active PIC or O/T_Suspended PIC, in order to avoid interference between call setup and user interaction.

During user interaction a "Mid-Call" event (EDP-R) can be detected, allowing the user to interrupt call processing and notify the SCF of this event. The SLP in the SCF may then either decide that:

– call processing can be resumed with a continue operation, i.e. the ongoing user interaction is unaffected; or

– a CPH or call processing operation other than continue is to be performed.

In the latter case the restriction applies that any ongoing user interaction shall be ended, i.e. the disconnect of the SRF resource is required in order to allow the CPH or other call processing operations (e.g. ReleaseCall) to be performed.

An SRF connection can be made while call processing is suspended, i.e. in response to a TDP-R or EDP-R, or when call processing is not suspended. All subsequent call processing operations and CPH operations received from SCF will not be executed until end of user interaction, with the exception of Continue/ContinueWithArgument operation. which is allowed also when call processing is suspended with user interaction ongoing. All operations leading to release of the Call Segment with SRF connection, such as ReleaseCall or MergeCallSegments will not be executed until end of user interaction on that Call segment (CP or leg).

The release of the Call Segment by any other entity than SCF, i.e. Abandon/Disconnect of last leg or the leg on which an SRF connection is made, will release the SRF on that Call Segment.

7.2.3.13 Handling of recorded voice messages

The PromptAndReceiveMessage is used for recording of messages like:

– personal greetings;

– voice messages;

– tone messages, etc.

When doing so it is possible by the SLP in the SCF in the PromptAndReceiveMessage to specify within 'InformationToRecord' a "MessageDeletionTimeOut" indicating the maximum time duration a message recording shall be stored in the SRF, i.e. to tell the SRF when to purge the recorded message. Therefore SCF can have control over recording and playback as well as deletion with this operation.

Moreover, some IPs will have the possibility to handle deletion directly between the user and the IP by means of e.g. DTMF.

Therefore, the capability exists for the SRF to purge the recorded message either via user interaction directly or via SCF control.

7.2.3.14 CSAID and its relation to Dialogue ID

- Dialogue ID

  The establishment of an INAP dialogue involves two application processes as described in Recommendation Q.1228, one that is the dialogue-initiator and one that is the dialogue-responder. On the functional plane a CallID Identifies a specific instance of a relationship between a SCF and SSF. At the physical plane for IN CS-2, it is mapped to a TCAP transaction identity or Dialogue ID.

- Created Call Segment Association ID

  This information element identifies for the SCF unambiguously the CSA instance in the SSF under control in the involved relationship instance. The SCF may use this information to address CSA instances in the SSF, for example when a call segment should be moved from one CSA instance under one SLPI control to another CSA instance in the same SSF under another SLPI control. The SSF is responsible for specifying a new CSA identifier for the created CSA which is unique within the SSF.

From the above it is clear that the Dialogue ID merely identifies a specific instance of a relationship between two FEs like e.g. between SCF and SSF, whereas the Created Call Segment Association ID (CSAID) identifies the CSA instance in the SSF under control by the SCF (SLPI) in the involved relationship instance identified (i.e. within the specific relationship instance identified by Dialogue ID).

There is one to one relationship between dialogueID and CSAID, i.e. for one dialogueID one CSAID.

ANNEX A

IN CS-2 service scenario examples

A.1 Example of the "User Interaction Script" concept: "Credit Card Calling" services

A.1.1 Assumptions

This subclause describes a "Credit Card Calling" service on an IN architecture based on the "User Interaction Script" concept, e.g. a share of service logic between the SCF and the SRF. The SCF runs the global service logic while the SRF runs the service logic dedicated to user interaction.

During the User Interaction, the SCF sends information (orders, additional information) to the SRF using the SCF-SRF operations: scriptRun, scriptInformation and scriptClose and the SRF sends back information (partial or final results, requirement for additional information) using the scriptEvent operation. These operations are correlated with each other using the script identifier called idS in this example.

The "User Interaction" script on the SRF is composed of several enhanced functions. One enhanced function running on the SRF is stopped each time a new one is launched by the SCF. For example, when the SCF asks databases for user authentication, the SRF plays a waiting music until the SCF requests it either to prompt the calling party of a wrong PIN number or to get a called party number. In the same way, on receipt of a release indication from the called party during the call setup (busy, no-answer, etc.), the SCF requests the SRF to prompt the calling party about his choice: release, follow-on, etc.

The SCF service logic could reach several states which are:

– **Authentication**: The SCF asks databases to authenticate the Credit Card Number and the PIN. Depending on the result, the following events may be encountered: Card OK, Card NOK (in this case the SCF manages the number of tries), "In service" card (people are using the card).

– **Call**: After having received the "Called Party Number", the SCF completes the call to the called party. During the "alerting" phase, the following event may be encountered: "Answer" receipt, NoAnswer (after temporization), Busy conditions. On receipt of such events, the SCF requests the SRF to prompt the calling party about his choice: release, follow-on, etc.

– **Comm**: During the active phase the user has reached his correspondent. The end of this state is: Hang Up of the called party, etc.

A.1.2 Enhanced functions on the SRF

The dedicated enhanced functions that compose the "Credit Card Calling" script on the SRF are:

AskCard: The purpose is to get the user card number. The SRF controls the waiting duration and the number of repetition of each prompt. The SRF has to collect the card number managing user attempt, controlling the data format, the cancellation, etc.

The possible exits are:

1) OK with card number; or

2) NOK with the cause (errors, cancellation, etc.).

In the first case SRF will stay on line and will entertain the user with music. In the second case the SRF will hang up after an information prompt like "Sorry, too many mistakes (response is NOK, cause = error num)".

AskTel: The SRF asks for the phone number.

Menu: The SRF prompts the user to know what to do. For example, in case of no answer, after a specific waiting time, the SRF prompts the calling party and asks the calling party if he wants to keep waiting or to stop alerting the called party. In the latter case, the SRF will ask the calling party if he wants to make a follow-on call, and then will get the new called party number.

WarningEnd: will prompt user of the near end of the call.

A.1.3 Message Sequence Chart

Figure A.1 illustrates the use of the "User Interaction Script" concept with one "Credit Card Calling" service. The enhanced functions defined in the previous subclause are invoked sequentially:

a) The SCF invokes the **askCard** function to authenticate the calling party and gets the User card number. The possible results are:

   1) The calling party has correctly dialled his card number.

   2) The calling party has not correctly dialled his card number.

b) If the calling party has correctly dialled his card number, then the SCF closes the **askCard** function **(1')** and invokes the **askTel** function **(1")** to get the Called Party Number. The possible results are:

   3) The calling party has not correctly dialled the Called Party Number.

   4) The calling party has correctly dialled the Called Party Number.

b') If the calling party has not correctly dialled his card number, then the SCF invokes the **askCard** function a second time **(2').**

c) If the calling party has correctly dialled the Called Party Number, then the SCF closes the **askTel** function **(3')** and completes the call to the called party. If the called party does not answer (expiry of the SCF "No-Answer" timer), the SCF invokes the **Menu** function **(5)** to propose to the calling party a Menu.

c') If the calling party has not correctly dialled the Called Party Number, then the SCF closes the **askTel** function **(3')** and User Interaction script.

d) The possible results are:

   6) The calling party wants to make a follow-on call but has not correctly dialled the Called Party Number.

   7) The calling party wants to make a follow-on call and has correctly dialled the Called Party Number.

   8) The calling party wants to continue alerting the called party.

e) If the calling party wants to make a follow-on call but has not correctly dialled the Called Party Number, then the SCF closes the **Menu** function and User Interaction script **(6')**.

e') If the calling party wants to make a follow-on call and has correctly dialled the Called Party Number, then the SCF closes the **Menu** function **(3')** and completes the call to the called party. On receipt of a release indication from the called party, the SCF invokes the **WarningEnd** function to release the call and the User Interaction is complete **(9).**

e") If the calling party wants to continue alerting the called party, on receipt of the "no-answer" indication from the SSF, the SCF invokes the **Menu** function a second time **(10)**.

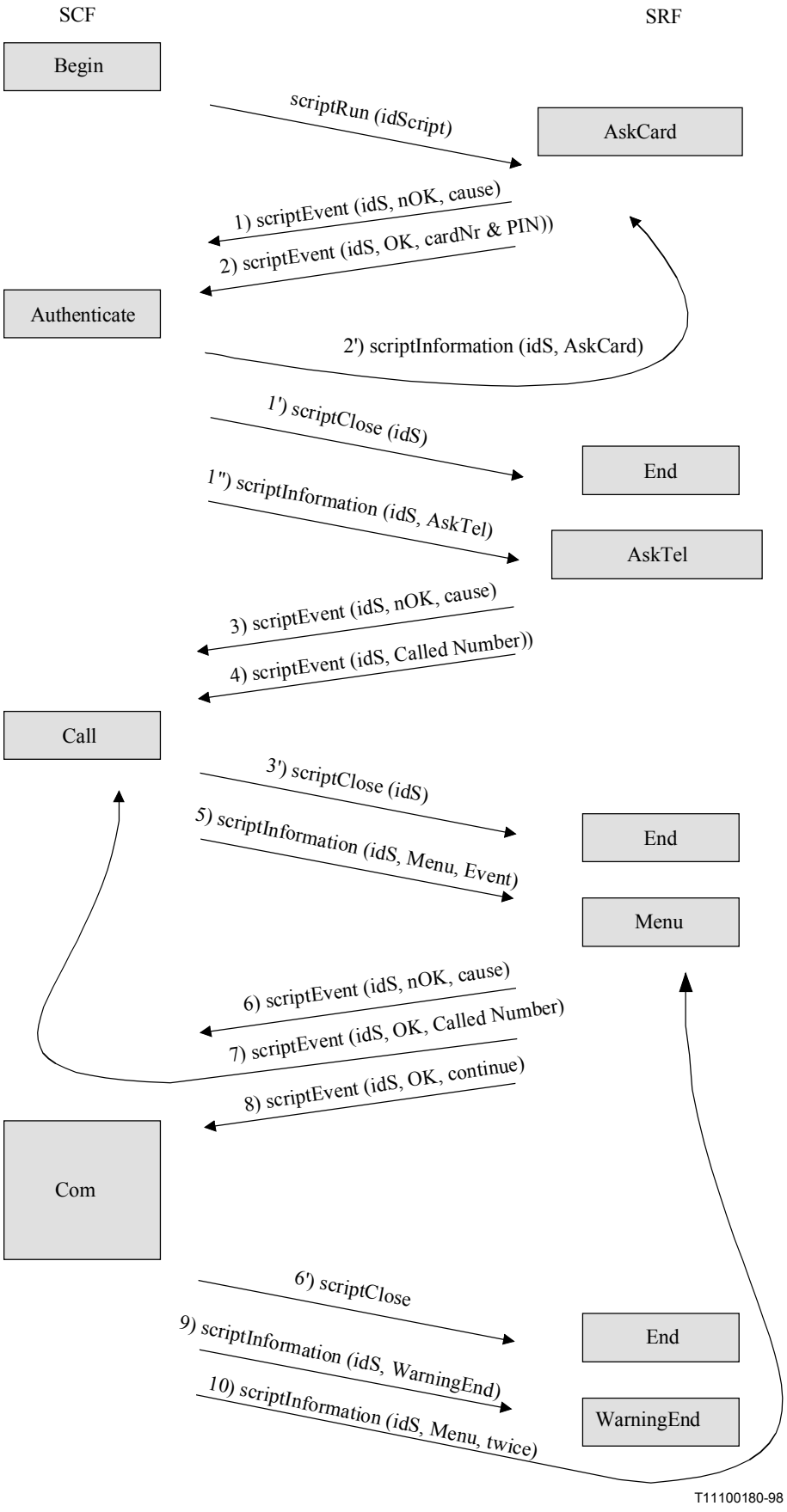


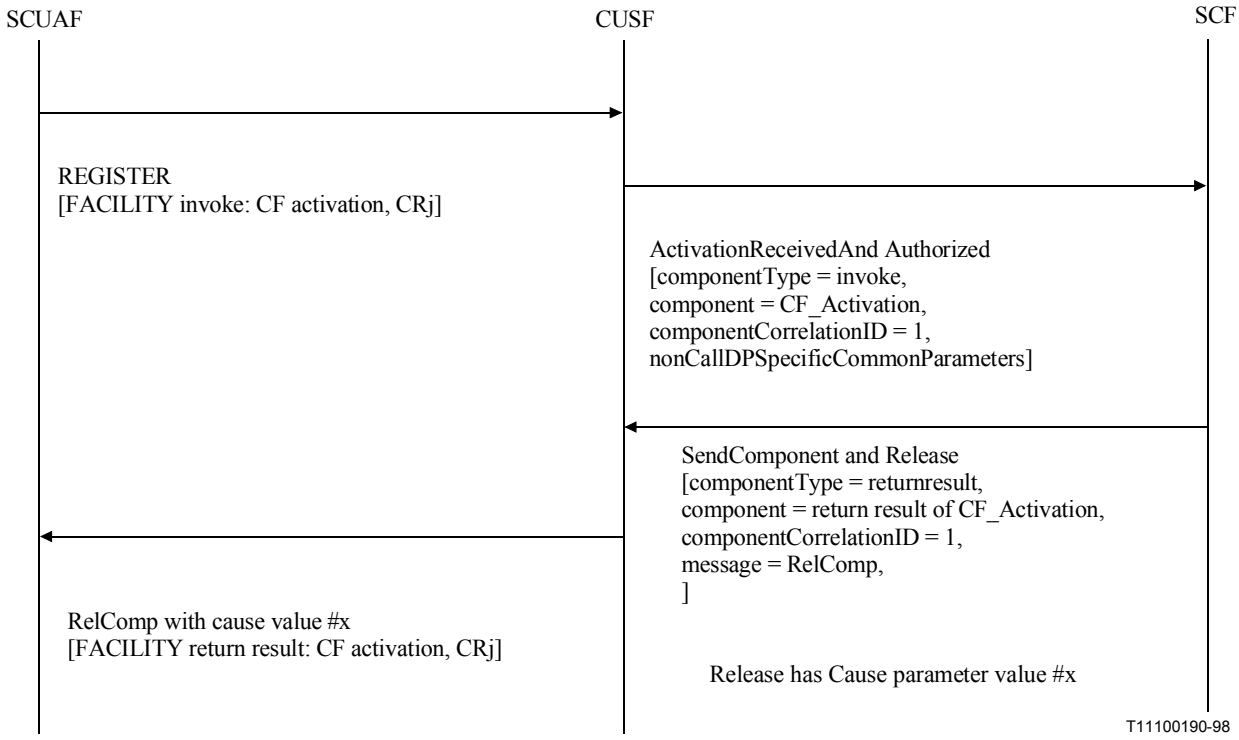
Figure A.1/Q.1229 – Example information flows for "Card Calling"

A.2 Service scenario examples for Out-channel Call Unrelated User Interaction

A.2.1 Call Forwarding activation request

The Message Sequence Chart (MSC) in Figure A.2 illustrates a simple case: the user activating Call Forwarding (CF) via the SCUAF. CF activation is identified at the CUSF by the operation code of FACILITY IE in the REGISTER message. Then the TDP criteria is checked, the CUSF issues ActivationReceivedAndAuthorized (ARAA) as a TDP-R message.

The componentType, component, componentCorrelationID in ARAA are key information to correlate the response from the SCF (in this case SendComponent). As the invocation on the UNI (SCUAF-CUSF relationship) has only local significance, the component on the UNI is mapped to componentType and component parameters and the invokeID on the UNI is also indirectly mapped to componentCorrelationID. The componentCorrelationID will be managed in the CUSF to correlate a resource from the SCF for the Classes 2, 3, and 4 ROSE operations on the UNI that resulted in this triggering. The ID is assigned at the FE which detects (CUSF case) or decides (via SendComponent from SCF case: this is mentioned in A.1.2) the new invocation of operation on the UNI.



NOTE – This example assumes CR management will be realized in the CCF, and feature interaction management will be realized in the CUSF.

Figure A.2/Q.1229 – Example Message Sequence Chart for Call Forwarding activation

A.2.2 Call Forwarding activation request with authentication

This MSC illustrates that the SCF side (an SLP) decided to have another invocation on the UNI. The correlation of the component for authentication will be also done by componentCorrelationID, but the value (0xff) is assigned by the SCF. The ID is also used to correlate the event report request of return result or return error (via RequestReportBCSMEvent) and the event report (ComponentReceived) in this case.

In the example in Figure A.3, the value space for the componentCorrelationID is divided in positive and negative to ease the management of the ID in the SCF and the CUSF respectively.

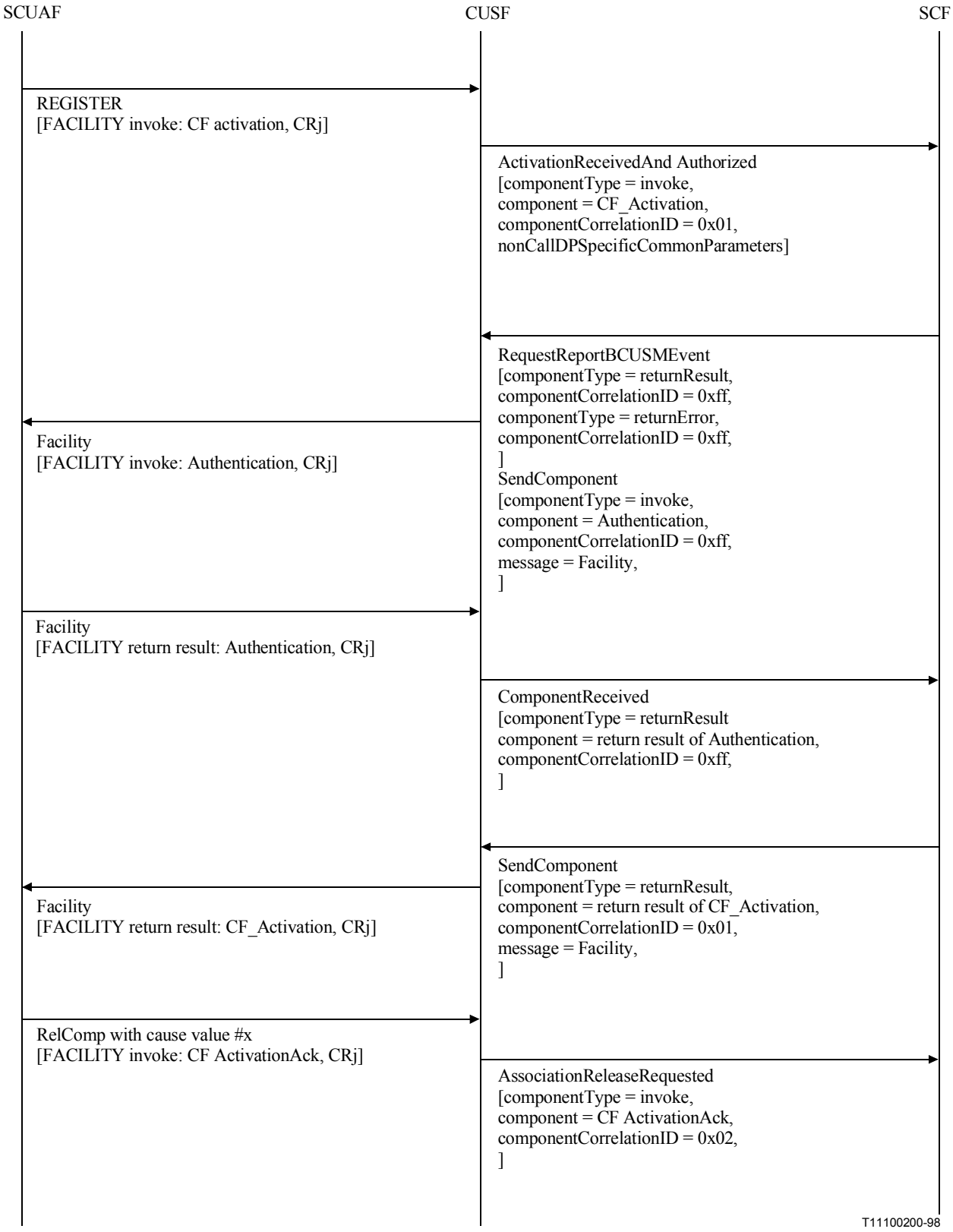


Figure A.3/Q.1229 – Example Message Sequence Chart for Call Forwarding activation (with authentication)

A.3 Service scenario examples for CPH CVS approach

A.3.1 Follow-on Call on request by calling party

This feature (see Figure A.4) allows a service user, e.g. a UPT user, when terminating an outgoing UPT call, before disconnecting completely, to initiate a new UPT outgoing call without having to repeat the identification and authentication procedures.

In the example in Figure A.5, during the alerting phase or the active call phase, the calling A-party requests the service logic to disconnect the connection from the SSP to the called B-party (outgoing call leg). The follow-on request from the user is considered as a "Mid-Call" event. After performing the disconnection, the service logic prompts the user via User Interaction procedures for the new address information to set up a new outgoing call.

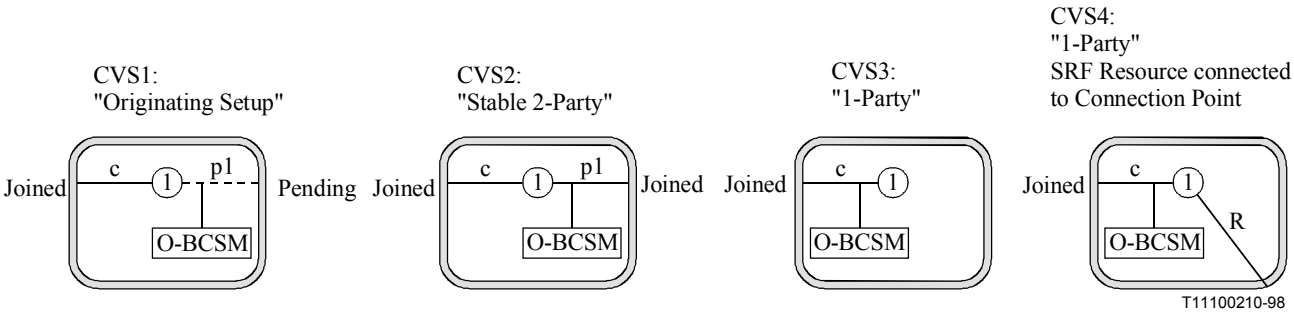


Figure A.4/Q.1229 – Graphical representation of CVSs used in "Follow-on Call"

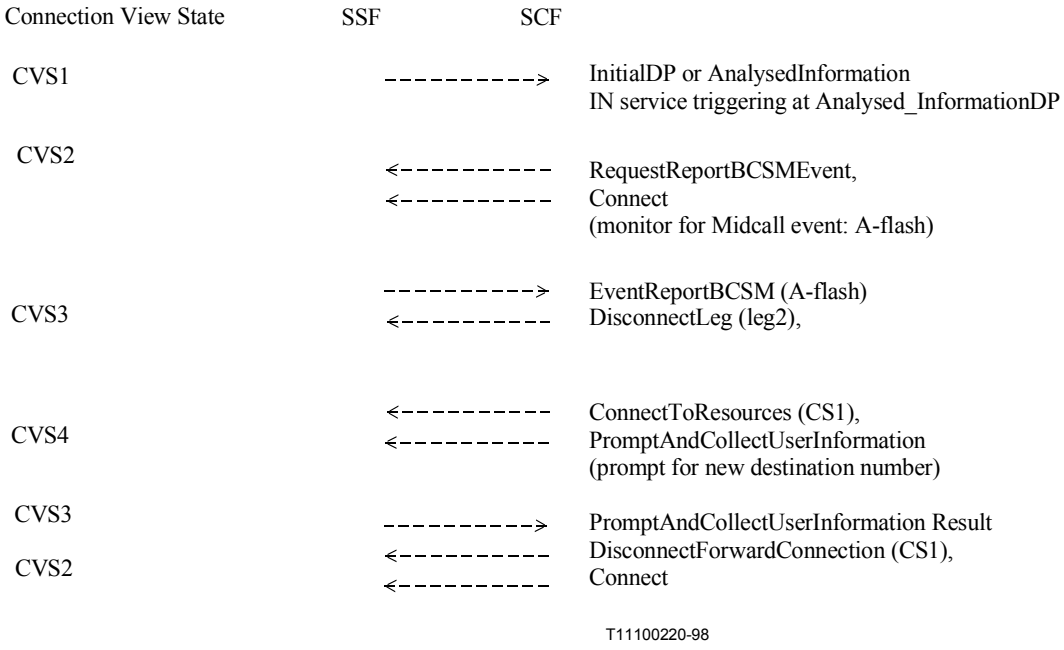


Figure A.5/Q.1229 – Example information flows for "Follow-on Call"

A.3.2 Reverse Charging

This service feature (see Figure A.6) allows the service subscriber (e.g. called party) to accept to receive calls at its expense and be charged for the entire cost of the call.

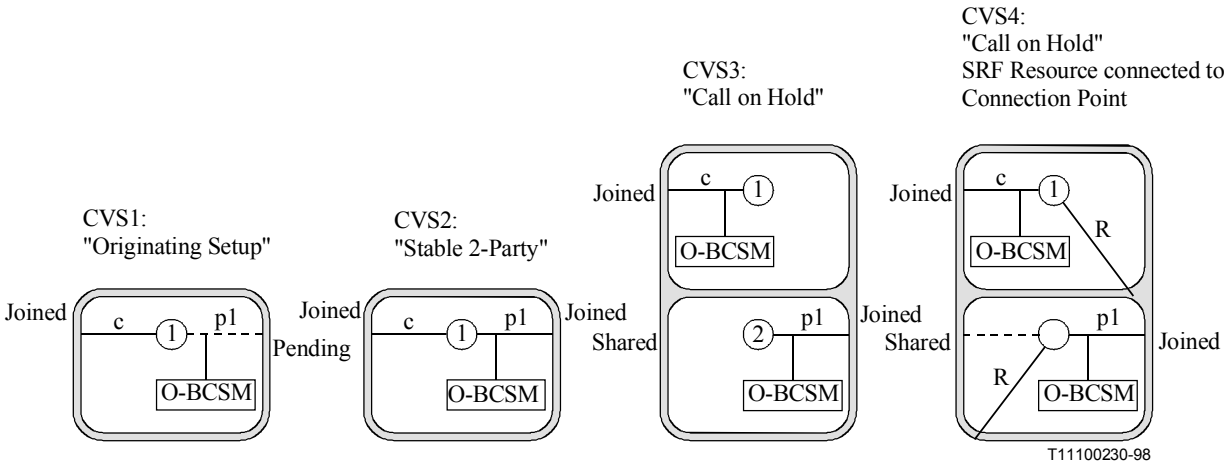


Figure A.6/Q.1229 – Graphical representation of CVSs used in "Reverse Charging"

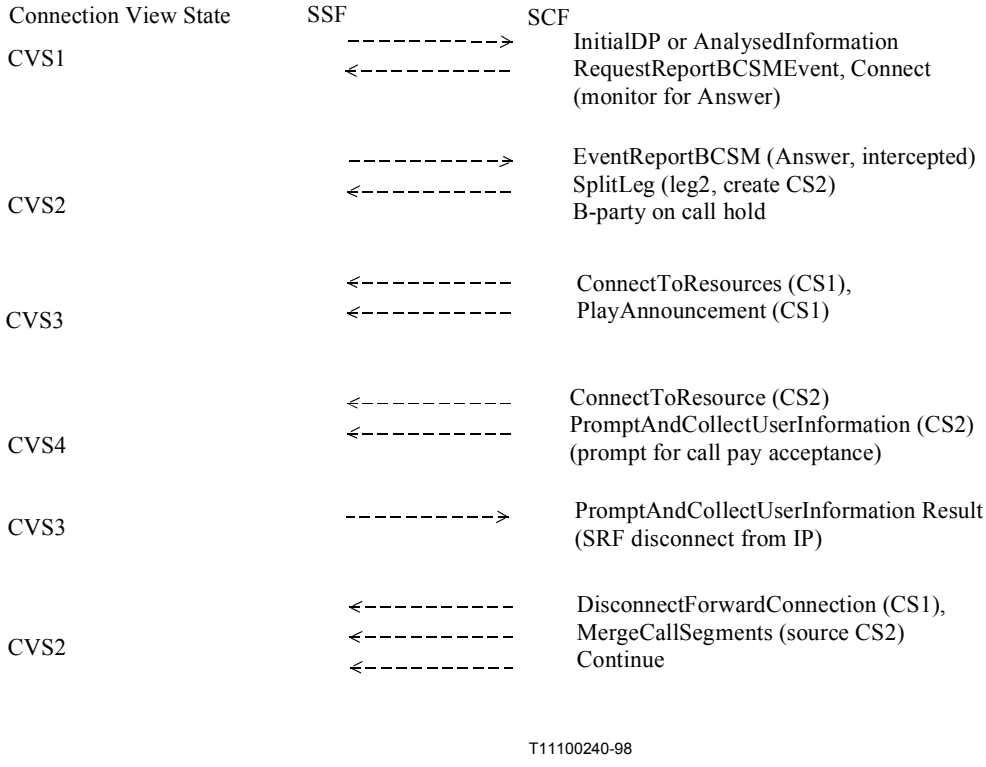


Figure A.7/Q.1229 – Example information flows for "Reverse Charging"

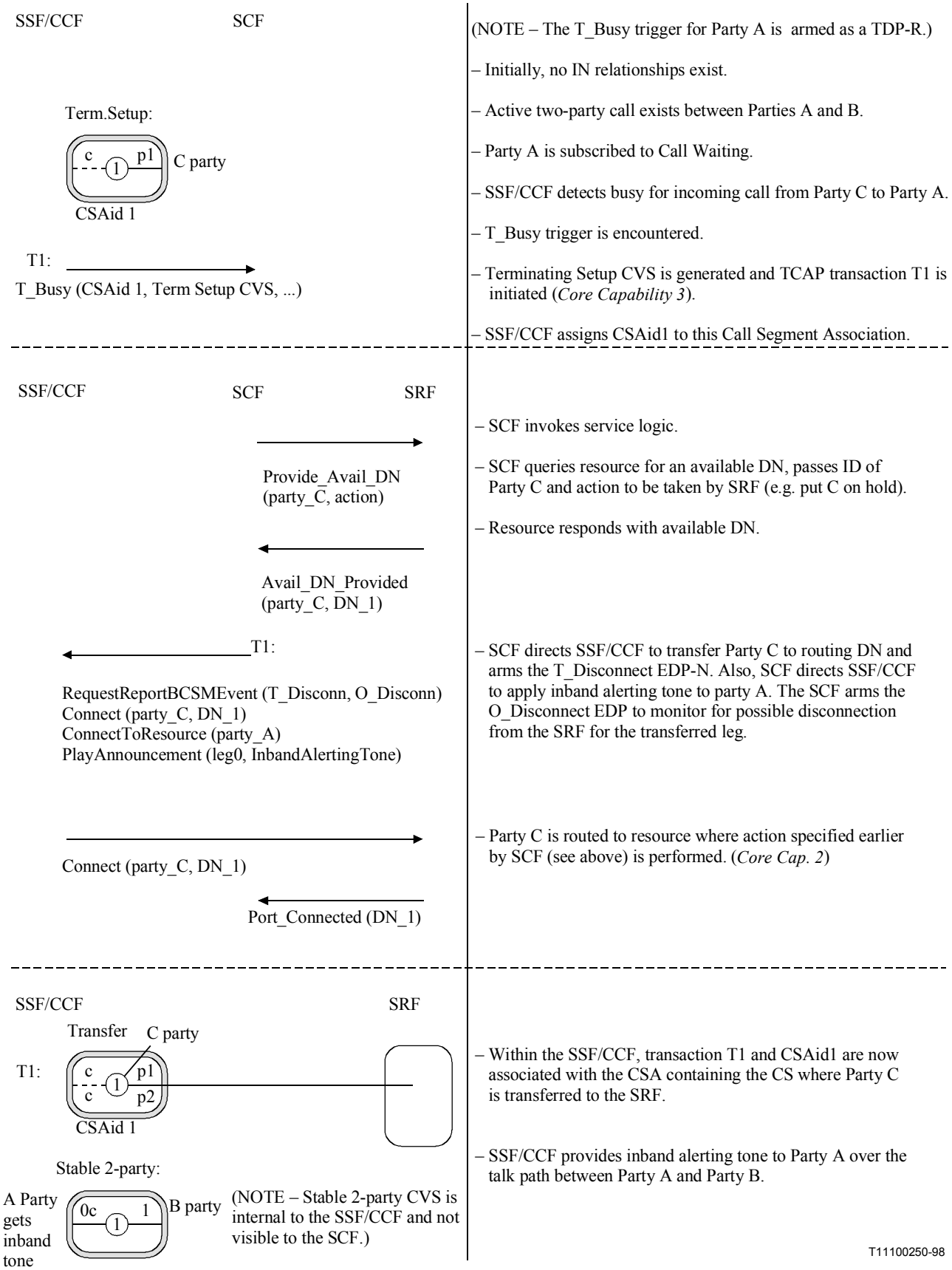
A.4 Service scenario examples for CPH hybrid approach

A.4.1 Call Waiting

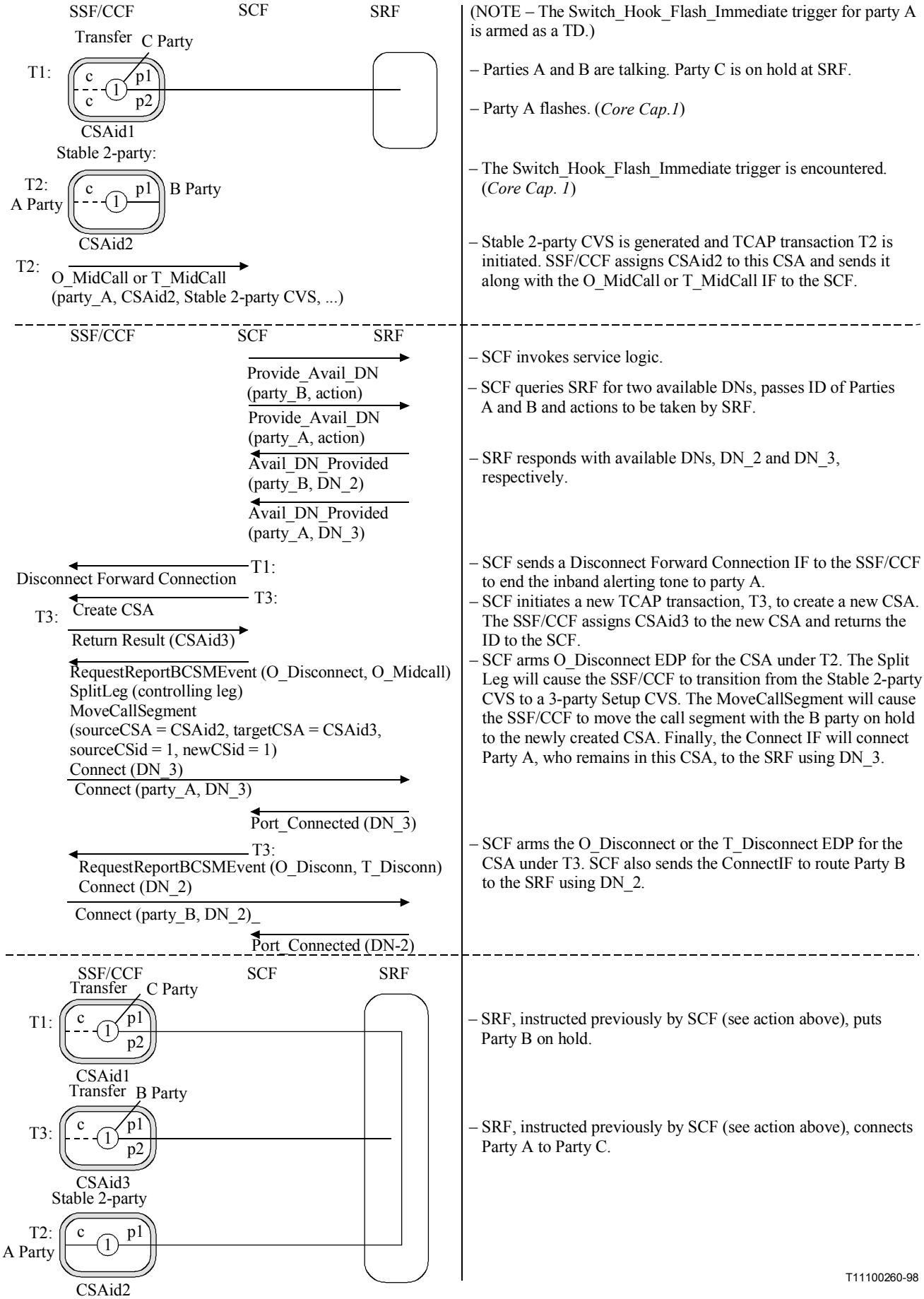
The following diagrams are used to illustrate how the Call Waiting feature can be implemented using the Call Party Handling (CPH) Hybrid Approach. The notation (*Core Cap. x*) indicates that a description corresponds to Core Capability *x*, where *x* = 1, 2, 3 or 4. The four Core Capabilities as identified in Recommendation Q.1224 are:

1) Core capability 1 allows for a user to enter information during a midcall event;
2) Core capability 2 is the ability of the SSF/CCF to connect a call party to an external resource to perform a transfer;
3) Core capability 3 is the ability of the SSF/CCF to present the current view of the call to the SCF;
4) Core capability 4 is the ability of the SSF/CCF to combine separate calls into a single call.

Call Waiting – Part 1
(Route Party C to Resource and Alert Party A of Second Incoming Call)



Call Waiting – Part 2
(Party A flashes to put Party B on hold, and to connect to Party C)



(NOTE – The Switch_Hook_Flash_Immediate trigger for party A is armed as a TD.)

– Parties A and B are talking. Party C is on hold at SRF.

– Party A flashes. (*Core Cap.1*)

– The Switch_Hook_Flash_Immediate trigger is encountered. (*Core Cap. 1*)

– Stable 2-party CVS is generated and TCAP transaction T2 is initiated. SSF/CCF assigns CSAid2 to this CSA and sends it along with the O_MidCall or T_MidCall IF to the SCF.

– SCF invokes service logic.

– SCF queries SRF for two available DNs, passes ID of Parties A and B and actions to be taken by SRF.

– SRF responds with available DNs, DN_2 and DN_3, respectively.

– SCF sends a Disconnect Forward Connection IF to the SSF/CCF to end the inband alerting tone to party A.

– SCF initiates a new TCAP transaction, T3, to create a new CSA. The SSF/CCF assigns CSAid3 to the new CSA and returns the ID to the SCF.

– SCF arms O_Disconnect EDP for the CSA under T2. The Split Leg will cause the SSF/CCF to transition from the Stable 2-party CVS to a 3-party Setup CVS. The MoveCallSegment will cause the SSF/CCF to move the call segment with the B party on hold to the newly created CSA. Finally, the Connect IF will connect Party A, who remains in this CSA, to the SRF using DN_3.

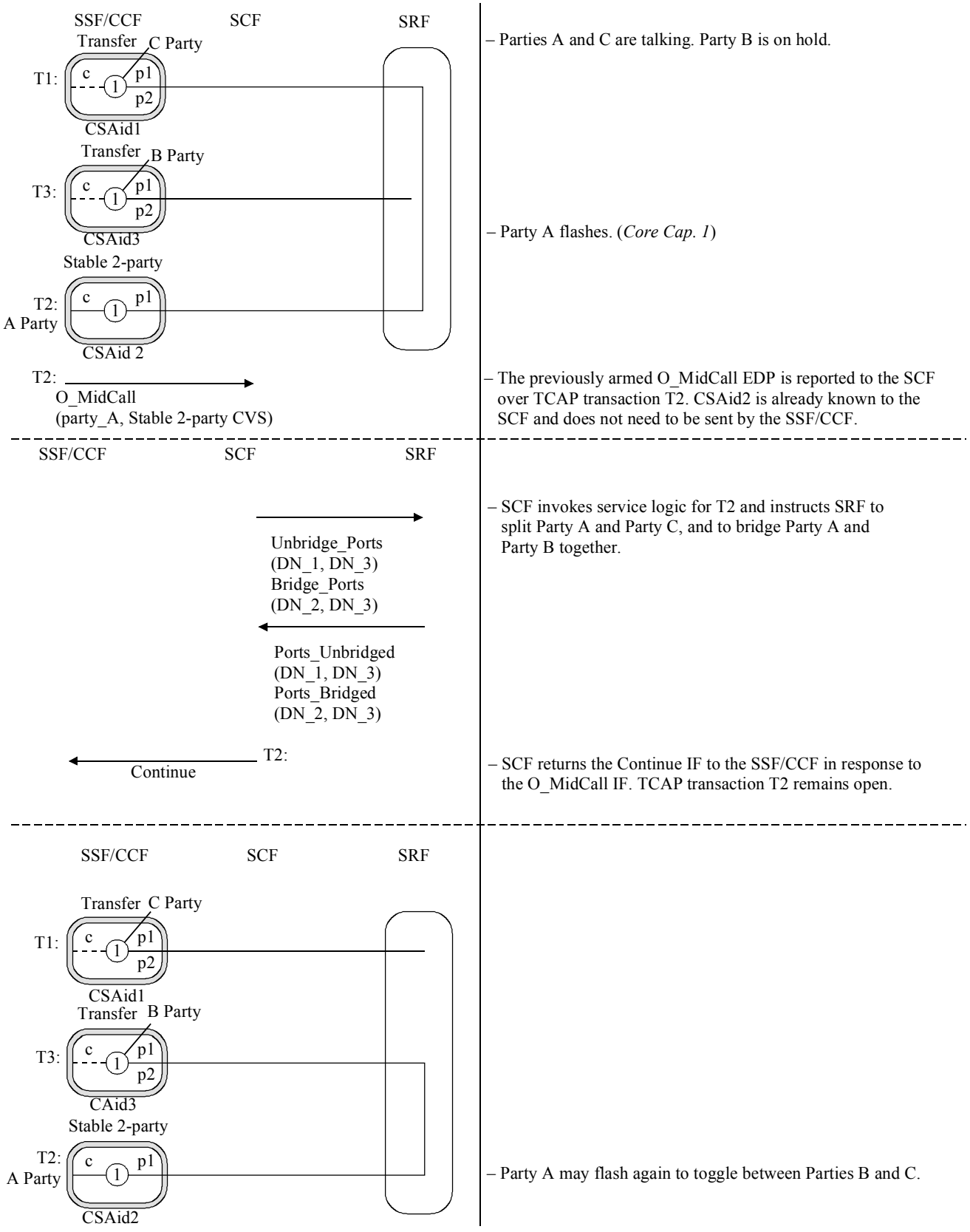
– SCF arms the O_Disconnect or the T_Disconnect EDP for the CSA under T3. SCF also sends the ConnectIF to route Party B to the SRF using DN_2.

– SRF, instructed previously by SCF (see action above), puts Party B on hold.

– SRF, instructed previously by SCF (see action above), connects Party A to Party C.

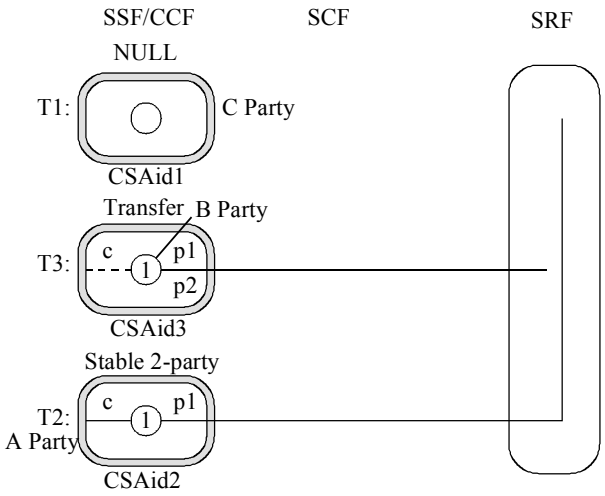
T11100260-98

Call Waiting – Part 3
(Party A flashes to put Party C on hold, and to connect to Party B)



T11100270-98

Call Waiting – Part 4
(Party C disconnects while talking to Party A, Party A is reconnected to Party B)



– Parties A and C are talking. Party B is on hold.

– Party C goes on hook.

T1:
T_Disconnect (leg = p1)

– SSF/CCF sends the T_Disconnect IF, using open TCAP transaction T1, to report the EDP-N event to the SCF.

Disconnect (Party C, DN_1)

Port_Disconnected (DN_1)

– SSF/CCF clears the call between Party C and the SRF and closes the transaction T1.

SSF/ CCF SCF SRF

T3:
Disconnect Leg (Leg id = 2)

– Having received the T_Disconnect IF over T1, the SCF understands there are only two parties left and decides to reconnect the parties back in the SSF/CCF. The SCF, using T3, instructs the SSF/CCF to disconnect party B from the SRF.

T2:
Disconnect Leg (Leg id = 1)
MoveCallSegment
(sourceCSA = CSAid3, targetCSA = CSAid2
sourceCSid = 1, newCSid = 2)
MergeCallSegments

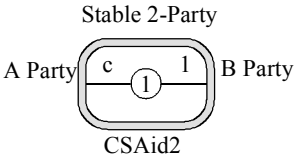
– The SCF, using T2, instructs the SSF/CCF to disconnect party A from the SRF. The SCF then requests the SSF/CCF to move the call segment from CSAid3 into this CSA, and then merge the two call segments together into a stable 2 party call between A and B.

Disconnect (Party A, DN_3)

Port_Disconnected (DN_3)

Disconnect (Party A, DN_2)

Port_Disconnected (DN_2)



– Both TCAP transactions, T2 and T3, are closed.

– The transaction between the SCF and SRF is closed.

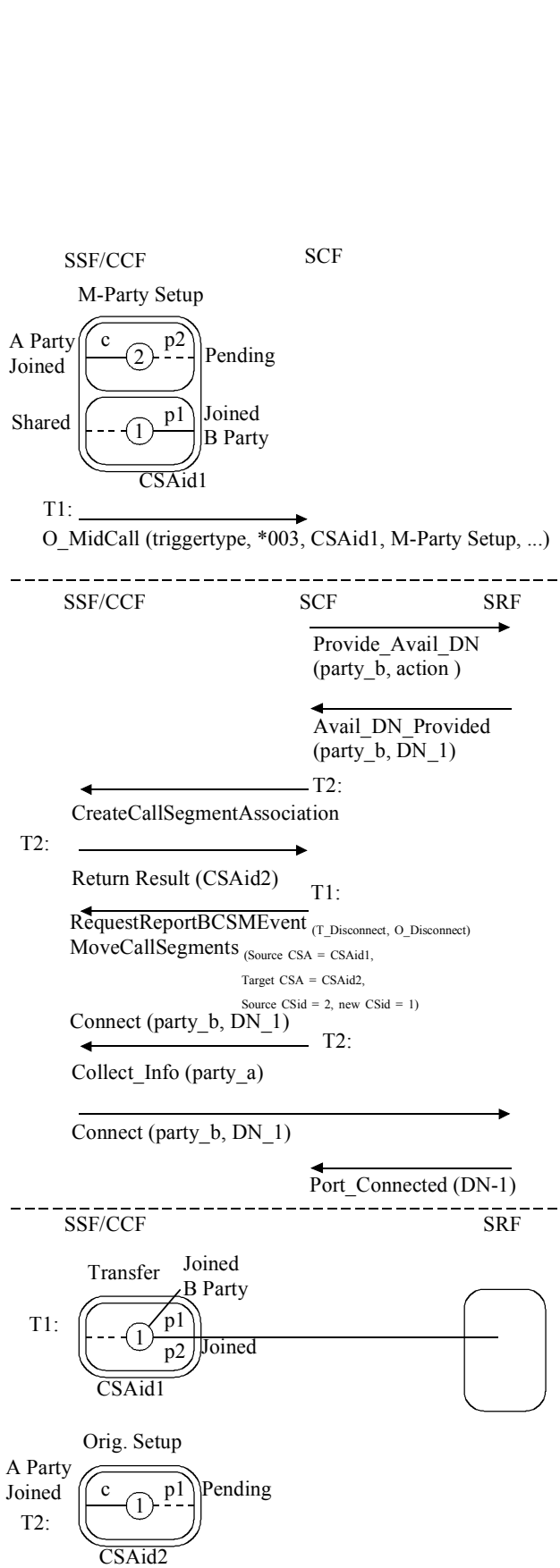
T11100280-98

A.4.2 Conference Call

The following diagrams are used to illustrate how the Conference Calling feature can be implemented using the Call Party Handling (CPH) Hybrid Approach. The notation (*Core Cap. x*) indicates that a description corresponds to Core Capability *x*, where *x* = 1, 2, 3 or 4. The four Core Capabilities as identified in Recommendation Q.1224 are:

1) Core capability 1 allows for a user to enter information during a midcall event;
2) Core capability 2 is the ability of the SSF/CCF to connect a call party to an external resource to perform a transfer;
3) Core capability 3 is the ability of the SSF/CCF to present the current view of the call to the SCF;
4) Core capability 4 is the ability of the SSF/CCF to combine separate calls into a single call.

Conference Call – Part 1 (Route Party B to Resource)



– Initially, no IN relationships exist.

– Active two-party call exists between Parties A and B.

– Call Party A flashes. (*Core Cap. 1*)

– B party placed on SSF/CCF based hold. (*Core Cap. 1*)

– SSF/CCF provides dial tone to A Party. (*Core Cap. 1*)

– Digit collector is connected to A Party. (*Core Cap. 1*)

– Party A: Enters feature code (e.g. *003). (*Core Cap. 1*)

– O_MidCall trigger fires. (*Core Cap. 1*)

– M Party Setup CVS is generated and TCAP transaction T1 is initiated. (*Core Cap. 3*)

– Input of DN instead of feature code by Party A would have resulted in a call being originated, without firing the trigger.

– SCF invokes service logic based on feature code (e.g. *003).

– SCF queries resource for an available DN, passes ID of Party B and action to be taken by SRF.

– Resource responds with available DN.

– The SCF initiates a new TCAP transaction, T2, to create a new CSA. The SSF/CCF assigns CSAid2 to the new CSA and returns the ID to the SCF in the Return Result.

– Under TCAP transaction T1, the Move Call Segments moves Party A to the new CSA, disassociating Parties A and B. Request ReportBCSMEvent sets the O/T Disconnect EDPs to monitor B-party disconnect or disconnection of the transferred leg from the SRF (e.g. resulting from a problem). Connect transfers Party B in CSAid1 to the SRF using DN_1 and transits to the Transfer CVS. (*Core Cap. 2*)

– Collect_Info is sent on transaction T2 (Party A).

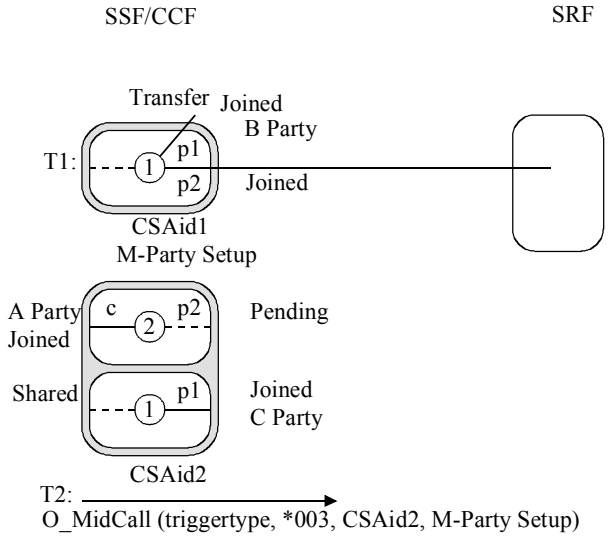
– Within the SSF/CCF, transaction T1 is now associated with Party B's path through the SSF/CCF (CSAid1) and T2 is associated with Party A (CSAid2).

– Party B is transferred to resource where action specified earlier by SCF (see above) is performed. (*Core Cap. 2*)

– Party A is given dial tone. Party A enters DN of Party C to originate a call to Party C. Internal to the SSF/CCF the Orig. Setup CVS for Party A will transit to the Stable 2-party CVS.

T11100290-98

Conference Call – Part 2 (Route Party C to Resource)



– Active two-party call is established between Parties A and C.

– Party B is still routed through SSF/CCF to resource. Within SSF/CCF, this path is associated with transaction T1.

– Call Party A flashes. (*Core Cap. 1*)

– C Party placed on SSF/CCF based hold. Internal to the SSF/CCF, this causes a CVS transition to M_Party Setup.

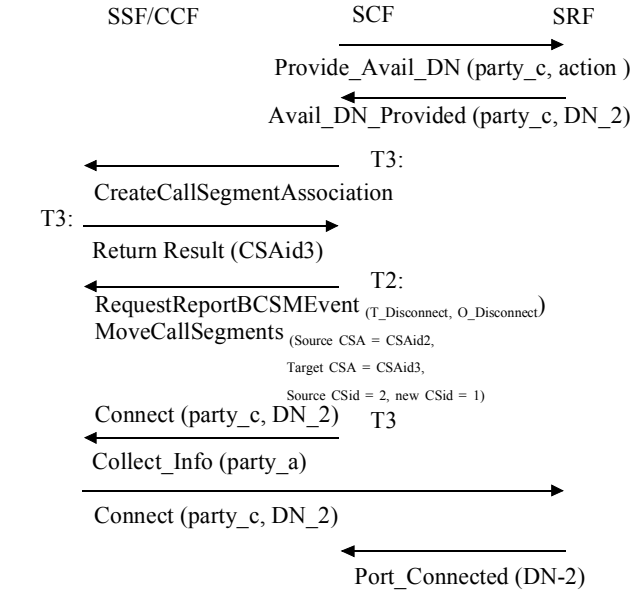
– Dial tone is provided to A Party. (*Core Cap. 1*)

– Digit collector is connected to A Party. (*Core Cap. 1*)

– Party A: Enters feature code (e.g. *003). (*Core Cap. 1*)

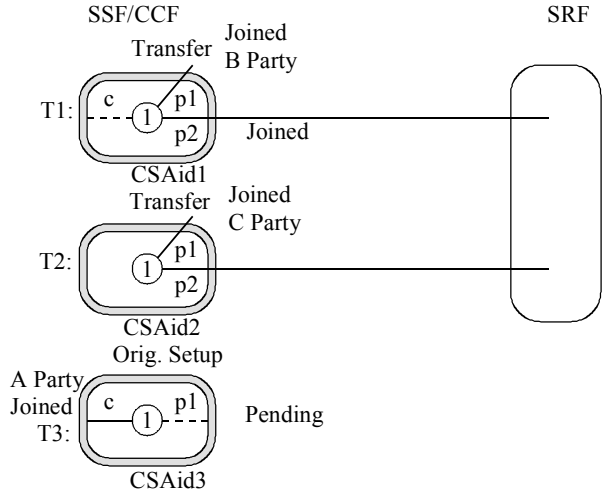
– O_MidCall trigger fires. M Party Setup CVS is sent to the SCF on transaction T2. (*Core Cap. 3*)

– Input of DN instead of feature code by Party A would have resulted in a call being originated, without firing the trigger.



– SCF invokes service logic based on feature code (e.g. *003).
– SCF queries resource for an available DN, passes ID of Party C and action to be taken by SRF.
– Resource responds with available DN.

– The SCF initiates a new TCAP transaction, T3, to create a new CSA. The SSF/CCF assigns CSAid3 to the new CSA and returns the ID to the SCF in the Return Result.
– Under TCAP transaction T2, the Move Call Segments moves Party A to the new CSA, disassociating Parties A and B. Request ReportBCSMEvent sets the O/T Disconnect EDPs to monitor C-party disconnect or disconnection of the transferred leg from the SRF (e.g. resulting from a problem). Connect transfers Party C in CSAid2 to the SRF using DN_2 and transits to the Transfer CVS. (*Core Cap. 2*)
– Collect_Info is sent on transaction T3 (Party A).
– Within the SSF/CCF, transaction T1 is now associated with Party B's path through the SSF/CCF (i.e. CSAid1), T2 is associated with party C (i.e. CSAid2) and T3 is associated with Party A (i.e. CSAid3).

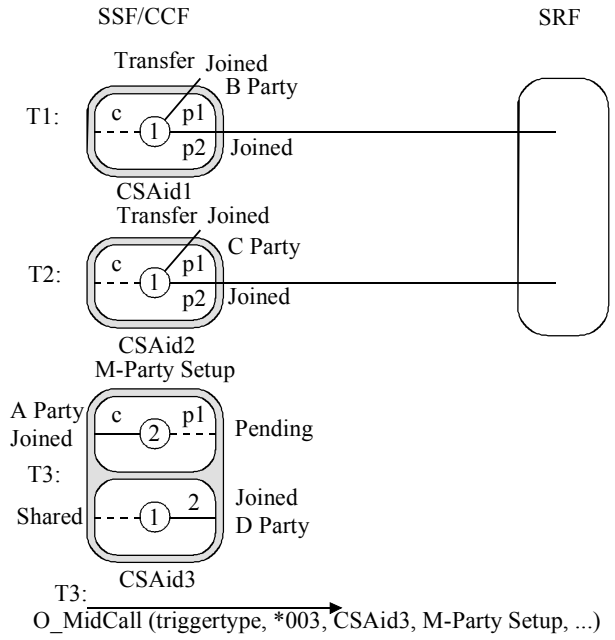


– Party C is routed to resource where action specified earlier by SCF (see above) is performed. (*Core Cap. 2*)

– Party A is given dial tone. Party A enters DN of Party D to originate a call to Party D. Internal to the SSF/CCF the Orig. Setup CVS for Party A will transit to the Stable 2-party CVS.

T11100300-98

Conference Call – Part 3 (Route Party D to Resource)



– Active two-party call is established between Parties A and D.

– Parties B and C are still routed through SSF/CCF to resource. Within SSF/CCF, these paths are associated with transactions T1 and T2.

– Call Party A flashes. (*Core Cap. 1*)

– D party placed on SSF/CCF based hold. Internal to the SSF/CCF, this causes a CVS transition to M_Party Setup. (*Core Cap. 1*)

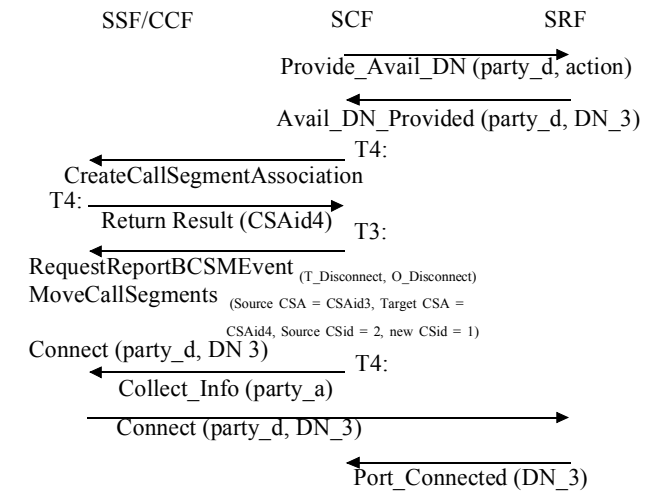
– Dial tone is provided to A Party. (*Core Cap. 1*)

– Digit collector is connected to A Party. (*Core Cap. 1*)

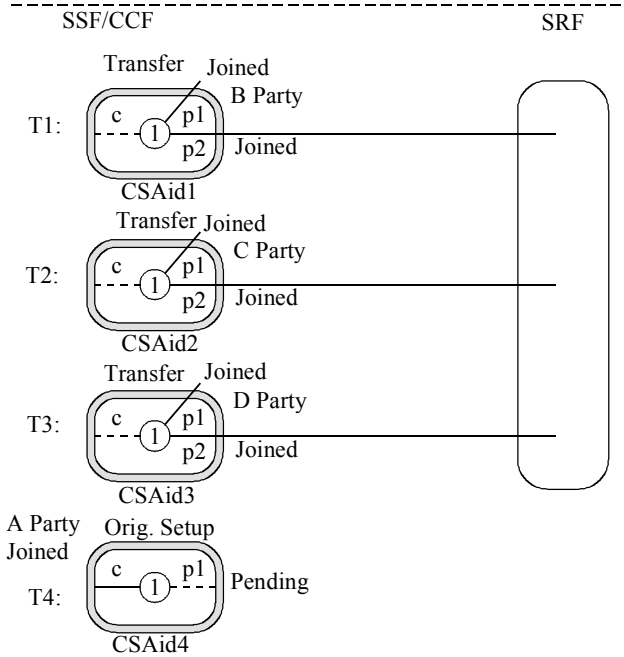
– Party A: Enters feature code (e.g. *003). (*Core Cap. 1*)

– O_MidCall trigger fires. M-Party Setup CVS is sent to the SCF on transaction T3. (*Core Cap. 3*)

– Input of DN instead of feature code by Party A would have resulted in a call being originated, without firing the trigger.



– SCF invokes service logic based on feature code (e.g. *003).
– SCF queries resource for an available DN, passes ID of Party D and action to be taken by SRF.
– Resource responds with available DN.
– The SCF initiates a new TCAP transaction, T4, to create a new CSA. The SSF/CCF assigns CSAid4 to the new CSA and returns the ID to the SCF in the Return Result.
– Under TCAP transaction T3, the Move Call Segments moves Party A to the new CSA, disassociating Parties A and D. RequestReportBCSMEvent sets the O/T Disconnect EDPs to monitor D-party disconnect or disconnection of the transferred leg from the SRF (e.g. resulting from a problem). Connect transfers Party D in CSAid3 to the SRF using DN_3 and transits to the Transfer CVS. (*Core Cap. 2*)
– Collect_Info is sent on transaction T4 (Party A).
– Within the SSF/CCF, transaction T1 is now associated with Party B's path through the SSF/CCF (i.e. CSAid1), T2 is associated with Party C (i.e. CSAid2), T3 with Party D (CSAid3) and T4 with Party A (e.g. CSAid4).

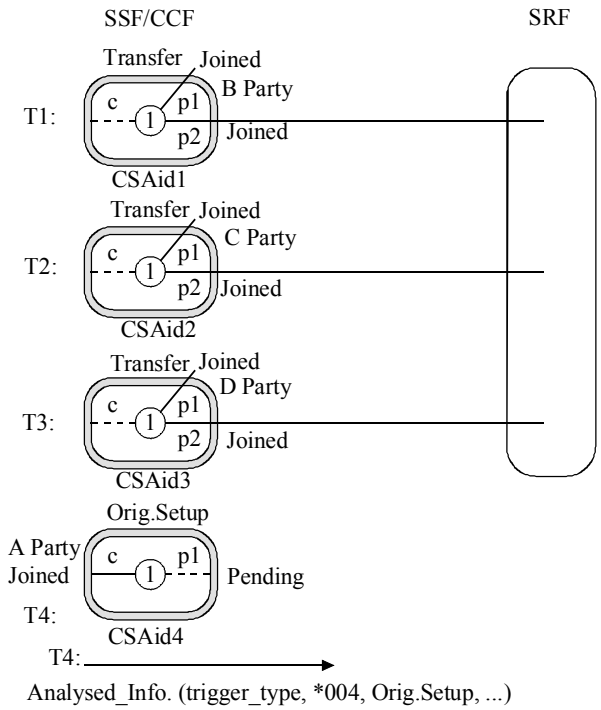


– Party D is routed to resource where action specified earlier by SCF (see above) is performed. (*Core Cap. 2*)

– Party A is given dial tone.

T11100310-98

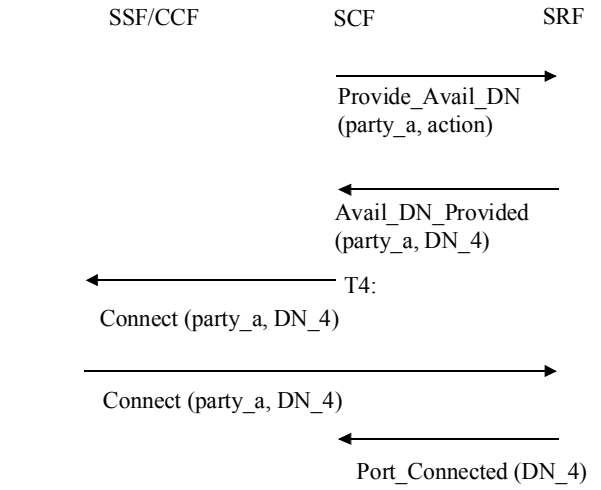
Conference Call – Part 4 (Route Party A to Resource)



– Parties B, C and D are still routed through SSF/CCF to resource. Within the SSF/CCF, these paths are associated with transactions T1, T2 and T3.

– Call Party A enters feature code (e.g. *004) to include itself in the conference call.

– Orig. Setup CVS is sent to SCF from Analysed_Info DP on Transaction T4. *(Core Cap. 3)*



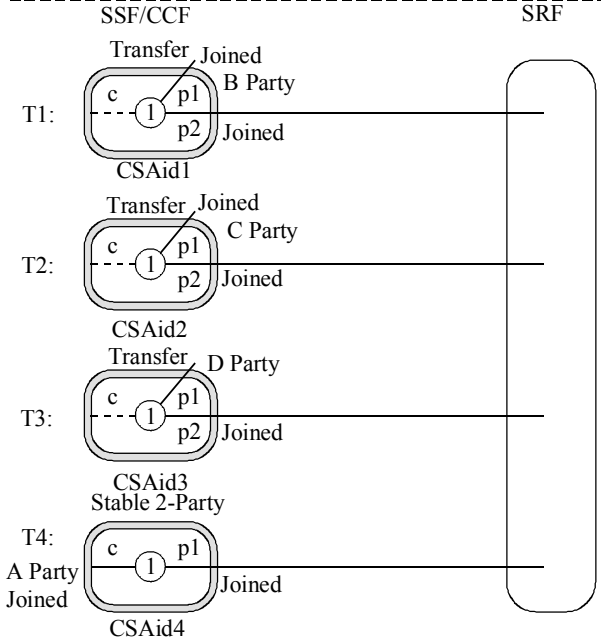
– SCF invokes service logic based on feature code (e.g. *004).

– SCF queries resource for an available DN, passes ID of Party A and action to be taken by SRF.

– Resource responds with available DN.

– SCF directs SSF/CCF over Transaction T4 to connect Party A to the SRF (i.e. routing DN_4). (*Core Cap. 2*)

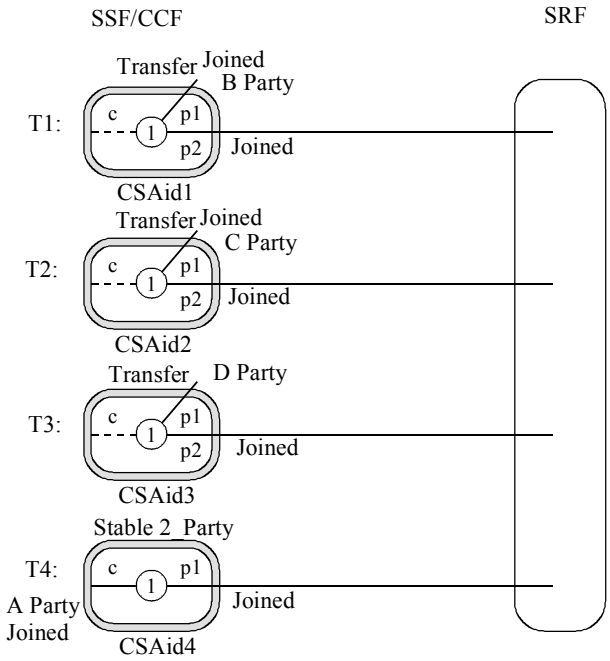
– Within the SSF/CCF, transaction T4 remains associated with Party A's path through the SSF/CCF.



– Party A is routed to resource where action specified earlier by SCF (see above) is performed. (*Core Cap. 2*)

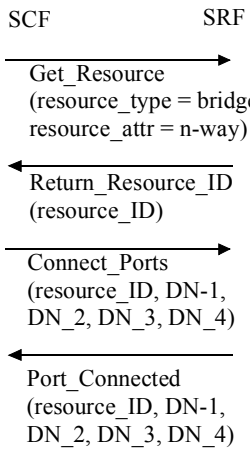
T11100320-98

Conference Call – Part 5 (Bridging at the Resource)

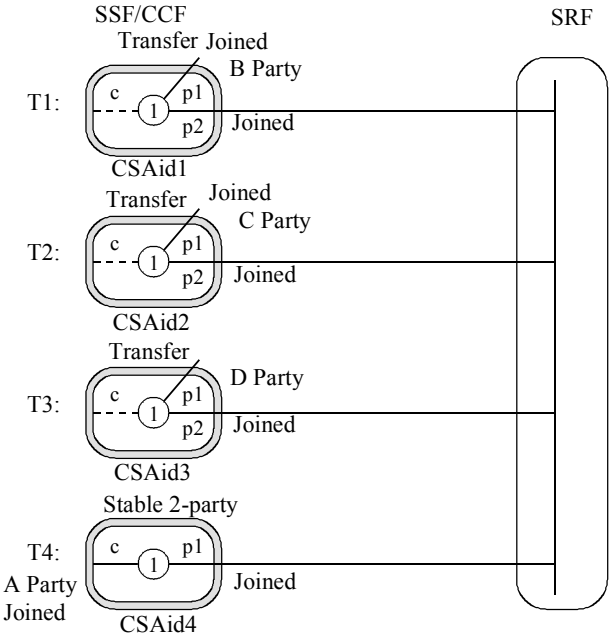


– Parties A, B, C and D have been routed to the resource.

– Party A is the controlling party.



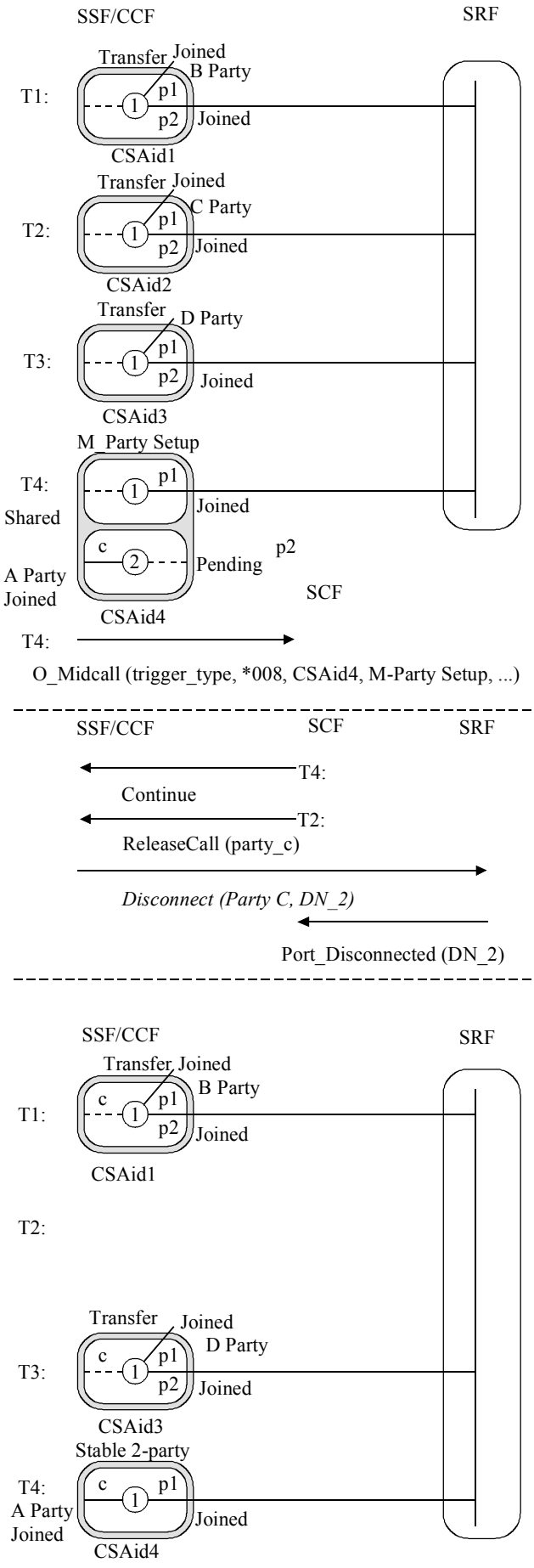
– SCF requests the use of an N-way bridging resource and that the four DNs used for Parties A, B, C and D be connected to the bridging resource.
– Connect_Ports can handle 1 to n ports.
– At any time, Party A disconnect can occur under the following conditions:
  • Party A goes on hook.
– At any time, Party B disconnect can occur under the following conditions:
  • Party B goes on hook;
  • Party A flashes and enters feature code (e.g. *007). (*Core Cap. 1*)
– At any time, Party C disconnect can occur under the following conditions:
  • Party C goes on hook;
  • Party A flashes and enters feature code (e.g. *008). (*Core Cap. 1*)
– At any time, Party D disconnect can occur under the following conditions:
  • Party D goes on hook;
  • Party A flashes and enters feature code (e.g. *009). (*Core Cap. 1*)



– Parties A, B, C and D are now connected in an N-way cell.

T11100330-98

Conference Call - Part 6 (Party Disconnect at the Resource, Initiated by Party A)



– Parties A, B, C and D are connected in an N-way call.

– Call Party A flashes. (*Core Cap. 1*)

– Party A's connection to resource is placed on SSF/CCF based hold. (*Core Cap. 1*)

– Dial tone is provided to A Party. (*Core Cap. 1*)

– Digit collector is connected to A Party. (*Core Cap. 1*)

– Party A enters feature code to disconnect Party C (e.g. *008). (*Core Cap. 1*)

– O_MidCall trigger fires (*Core Cap. 1*) and M-Party Setup CVS is sent to the SCF on transaction T4. (*Core Cap. 3*)

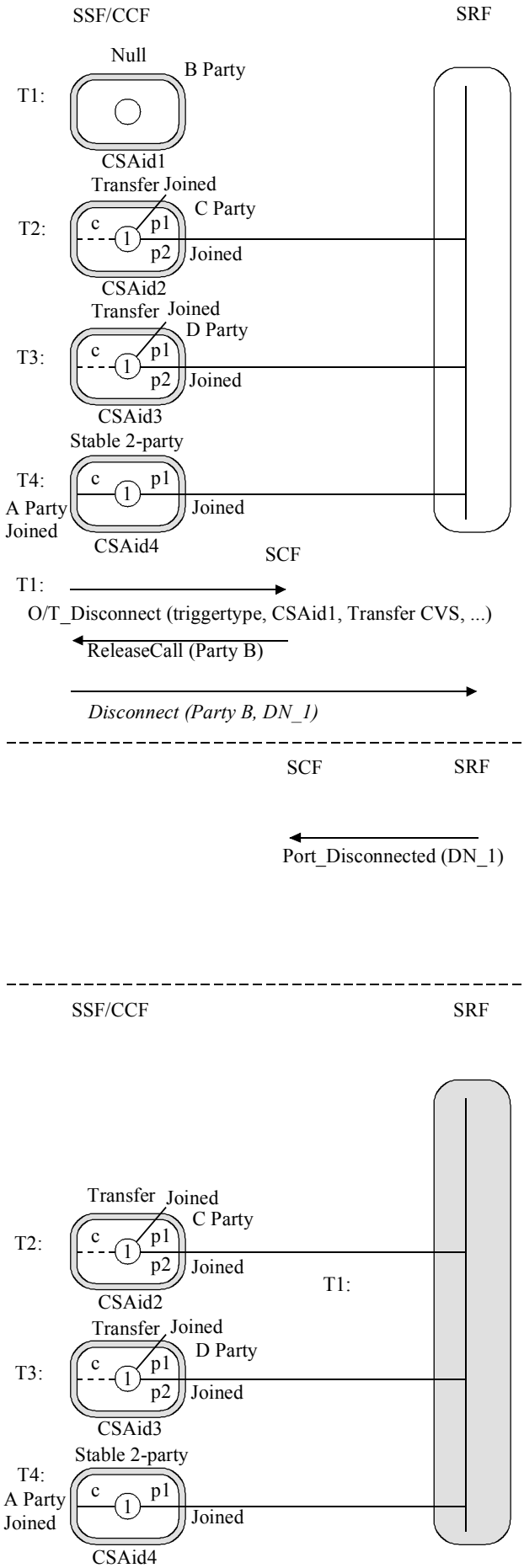
– Normal release of a connection applies. The SCF sends a ReleaseCall to the SSF/CCF on transaction T2 to release the call associated with Party C.

– T2 closes since no subsequent EDPs were set.

– Party A continues with the active call to the resource and is still communicating with Parties B and D.

T11100340-98

Conference Call – Part 7 (Party Disconnect at the Resource, Party B hangup)



– Parties A, B, C and D are connected in an N-way call.

– Call Party B goes on hook.

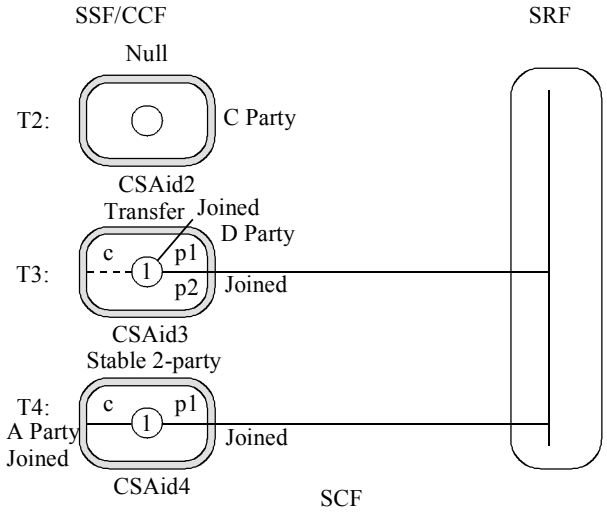
– O/T_Disconnect EDP fires and existing Transfer CVS for Party B is sent to the SCF on transaction T1. (*Core Cap. 3*). The SCF responds with a ReleaseCall to clear Party B from the conference and transit B to the Null CVS.

– Normal release of a connection applies.

– T1 closes since no subsequent EDPs were set.

T11100350-98

Conference Call – Part 8 (Re-establish Call at SSF/CCF)

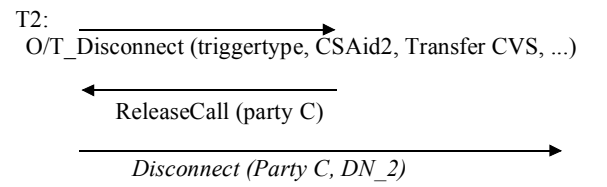


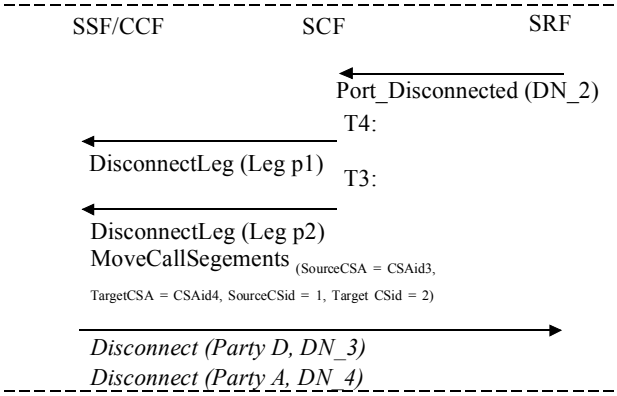
– Parties A, C and D are connected in an N-way call.

– Call Party C goes on hook.

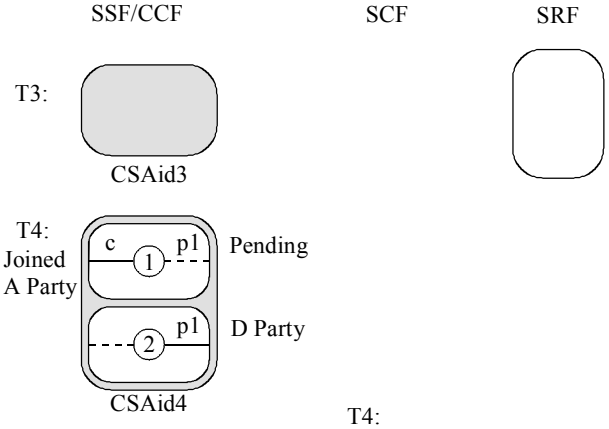
– O/T_Disconnect EDP fires and existing Transfer CVS for Party C is sent to the SCF on transaction T2. (*Core Cap. 3*). The SCF responds with a ReleaseCall to clear Party C from the conference and transit C to the Null CVS.

– Normal release of a connection applies to Party C.





– The SRF reports Party C's port is disconnected to the SRF. At this point the SCF understands that the bridging resource is only being used for a two-way connection between Parties A and D. Therefore, the SCF begins the process to terminate use of the resource and connect Parties A and D in the SSF.
– DisconnectLeg (p1) is sent on transaction T4 to disconnect A from the resource.
– DisconnectLeg (p2) is sent on transaction T3 to disconnect D from the resource.
– Party D's call segment (CSid1 inside CSAid3) is moved to CSAid4 and renumbered (CSid2). Party D is now associated with Party A in CSAid4. *(Core Cap. 4)*



– CSAid3 is deleted since it contains no further call segments.
– Transaction T3 is closed.

– MergeCallSegments is sent on transaction T4 to the SSF/CCF which then merges the calls for Parties A and D into a Stable 2-Party Call. *(Core Cap. 4)*

– The SCF initiates the release of the bridging resource when it is notified that no ports are making use of the resource.

– The transaction between the SCF and SRF is closed.

– Only transaction T4 remains open with Parties A and D in a Stable 2-party call in the SSF/CCF.

T11100360-98

A.4.3 Meet-Me Conference

The following diagrams are used to illustrate how the Meet-Me Conference feature can be implemented using the Call Party Handling (CPH) Hybrid Approach. The notation (*Core Cap. x*) indicates that a description corresponds to Core Capability *x*, where *x* = 1, 2, 3 or 4. The four Core Capabilities as identified in Recommendation Q.1224 are:

1) Core capability 1 allows for a user to enter information during a midcall event;

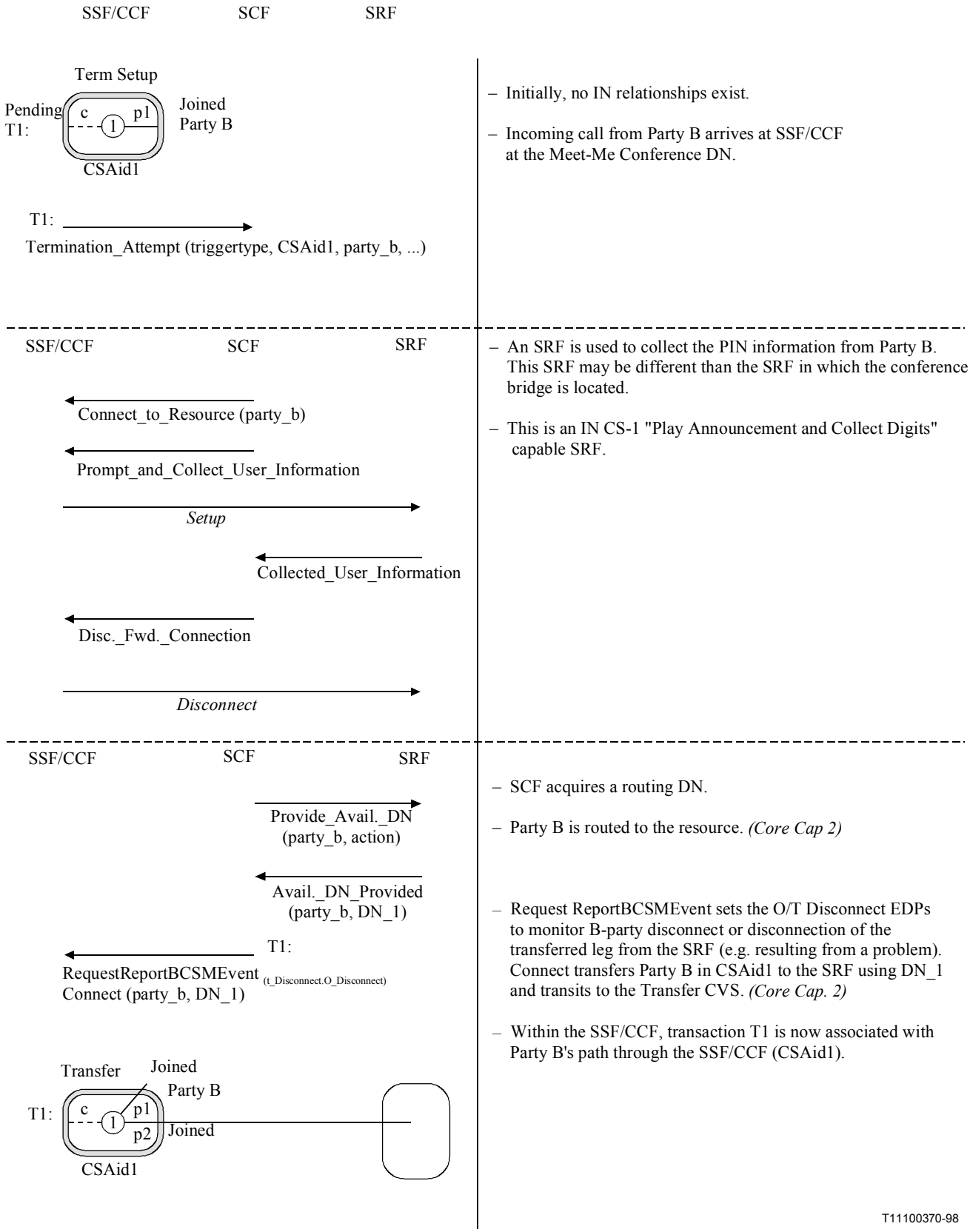
2) Core capability 2 is the ability of the SSF/CCF to connect a call party to an external resource to perform a transfer;

3) Core capability 3 is the ability of the SSF/CCF to present the current view of the call to the SCF;

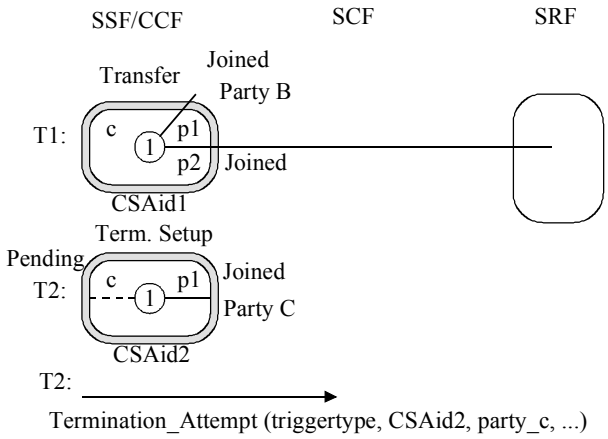
4) Core capability 4 is the ability of the SSF/CCF to combine separate calls into a single call.

These diagrams depict how parties are established in the Meet-Me conference and connected to the SRF. Once the conference parties are established, additional details as to manipulation of connections at the resource and the disconnect possibilities are equivalent to those that are illustrated in the previous scenarios for Conference Call (see A.4.2) and are therefore not repeated here.

Meet-Me Conference – Part 1 (Route Party B to Resource)

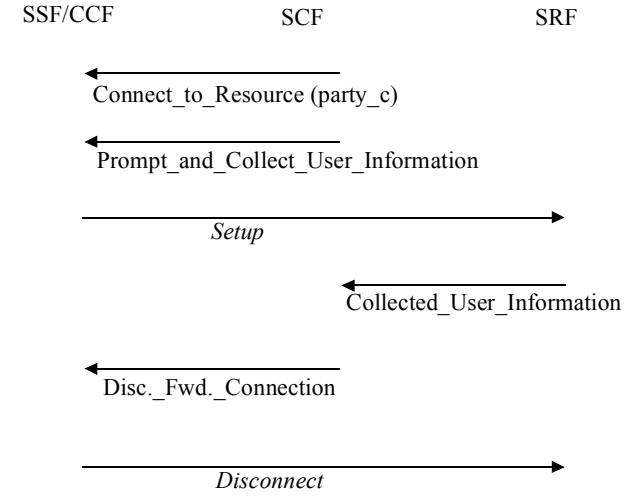


Meet-Me Conference – Part 2 (Route Party C to Resource)



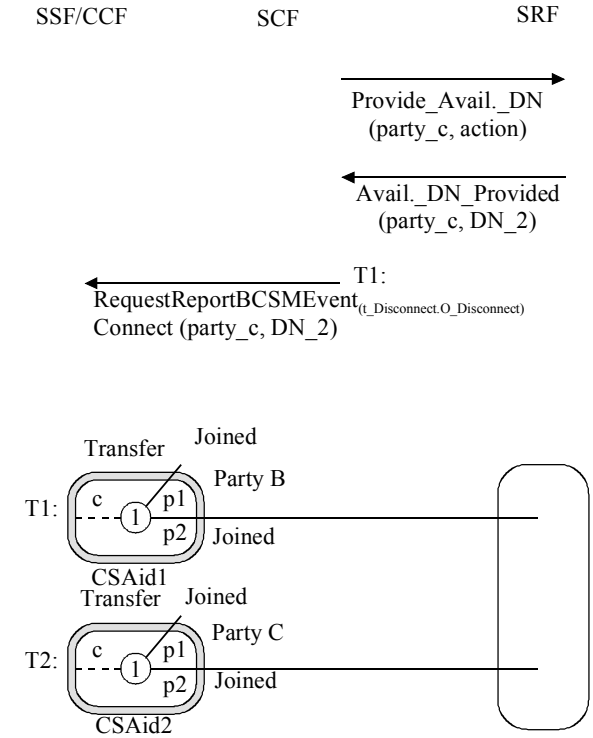
– Initially, no IN relationships exist.

– Incoming call from Party C arrives at SSF/CCF at the Meet-Me Conference DN.



– An SRF is used to collect the PIN information from Party C. This SRF may be different than the SRF in which the conference bridge is located.

– This is an IN CS-1 "Play Announcement and Collect Digits" capable SRF.



– SCF acquires a routing DN.

– Party C is routed to the resource. *(Core Cap. 2)*

– Request ReportBCSMEvent sets the O/T Disconnect EDPs to monitor C-party disconnect or disconnection of the transferred leg from the SRF (e.g., resulting from a problem). Connect transfers Party C in CSAid2 to the SRF using DN_2 and transits to the Transfer CVS. *(Core Cap. 2)*.

– Within the SSF/CCF, transaction T1 is now associated with Party B's path through the SSF/CCF (CSAid1) and T2 is associated with Party C.

T11100380-98

A.5 Internetwork Service Profile Transfer

A.5.1 Capability statement

Inter-Network Service Profile Transfer (ISPT) is a CS-2 telecommunications service feature which enables service profile information to be transferred to other service profile storage locations in other service providers. It is required to enable User Profile Information Portability.

A.5.2 Textual description

Consider three service providers, A, B and C in Figure A.8, who are cooperating to provide distributed mobility service.

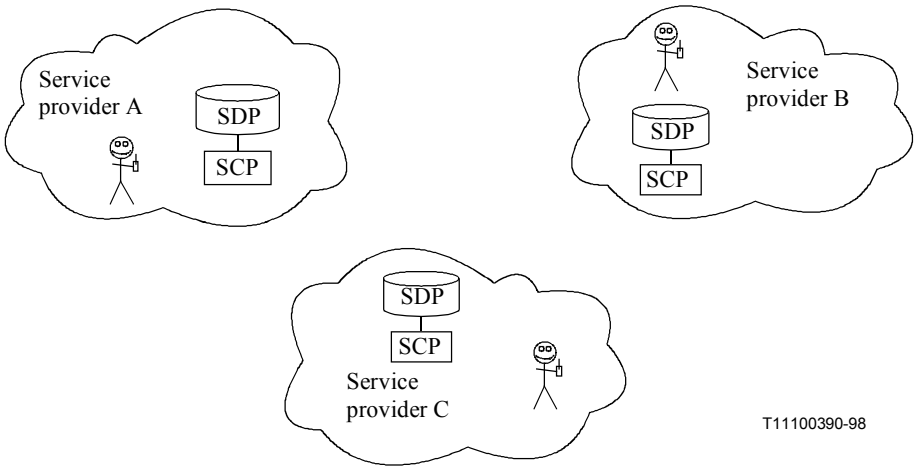


Figure A.8/Q.1229 – Mobility Service Provider scenario

In this example, each service provider has one SDP and has local mobility subscribers. Each SDP stores customer and service provider data, including mobility subscriber profiles.

A.5.2.1 Shadowing agreements

Service providers A and B have a requirement to share (shadow) roaming subscriber profiles. A shadowing agreement is an agreement to copy data between two SDPs, where one SDP holds the master data and the other SDP holds selected copies of the master data. As shown in Figure A.9, SDP A and B could maintain a shadowing agreement (shadowing agreement AB) for those users which are local to service provider A but can roam to service provider B. That is, when a user from service provider A roams to service provider B, SDP A will provide a shadow update, containing the roaming user's profile, to SDP B.

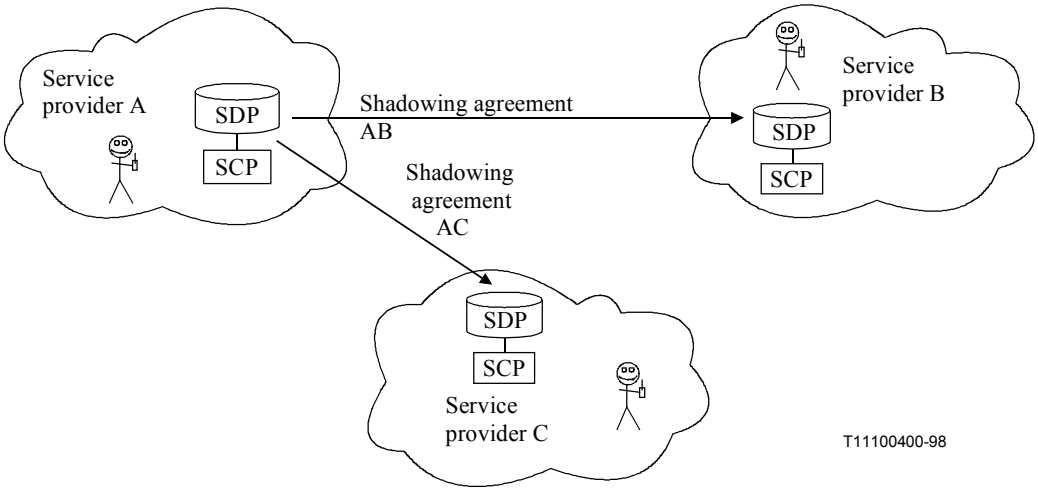


Figure A.9/Q.1229 – Shadowing agreements

A similar shadowing agreement (shadowing agreement BA) can be maintained for those instances where SDP B is the shadow supplier and SDP A is the shadow consumer. Shadowing agreements can also be arranged to support the cooperation of service providers A and C and service providers B and C.

A.5.2.2 Subscriber roams from home to visited service provider

Consider a subscriber whose service provider of choice is service provider A. As shown in Figure A.10, a user roaming into another service provider would result in the newly visited service provider receiving a copy of the roaming user's profile.

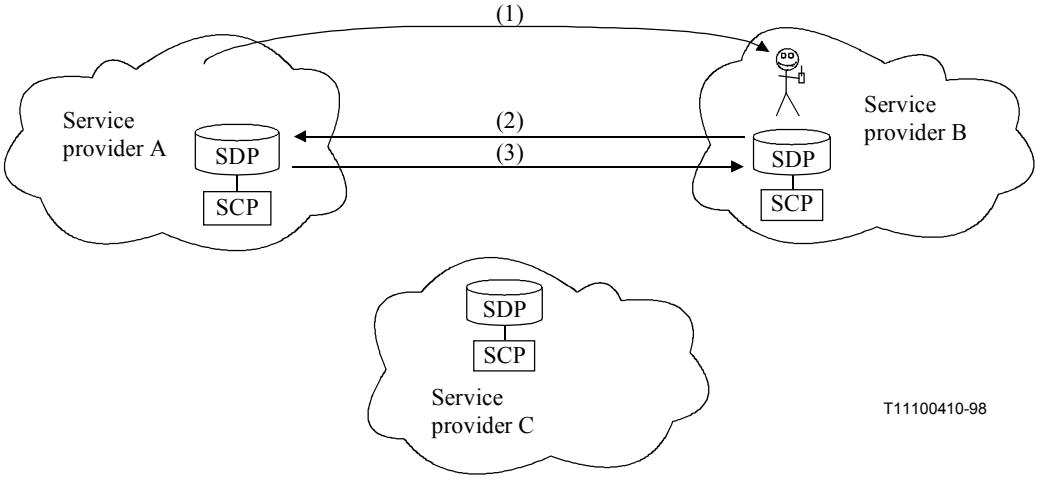


Figure A.10/Q.1229 – Subscriber roams from home to visited service provider

When the subscriber roams to service provider B, service provider B detects the presence of the roaming user. Service provider B then collects service and authentication information from the subscriber's equipment. From this information service provider B can determine the subscriber's unique identity and home service provider.

SCP B then modifies the visiting subscriber's profile, on the subscriber's home SDP, to indicate it as belonging to shadowing agreement AB. SDP A detects the change in its master information and sends a copy of the subscriber's profile to SDP B.

Figure A.11 illustrates the corresponding message sequence chart.

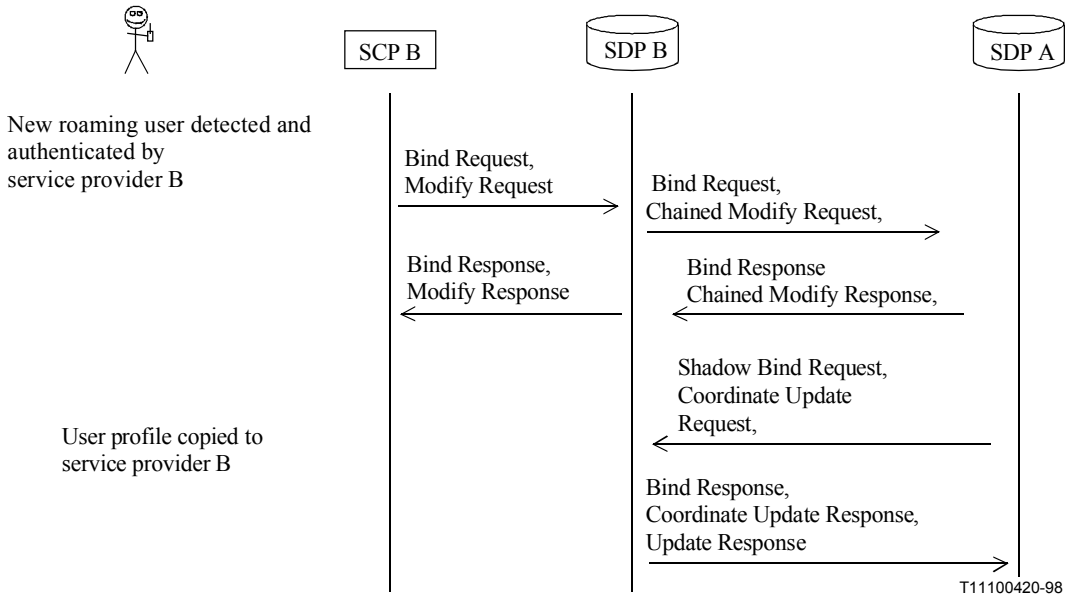


Figure A.11/Q.1229 – Home to visited service provider message sequence chart

Figure A.11 shows the case where directory operations are all sent in the same PDU and does not include message sequences for errors or for unbinding.

A.5.2.3 Subscriber roams from one visited service provider to another

Now suppose that the subscriber roams from service provider B to service provider C, as illustrated in Figure A.12. This would result in the copy of the roaming user's profile being removed from the previously visited service provider and being supplied to the newly visited service provider.

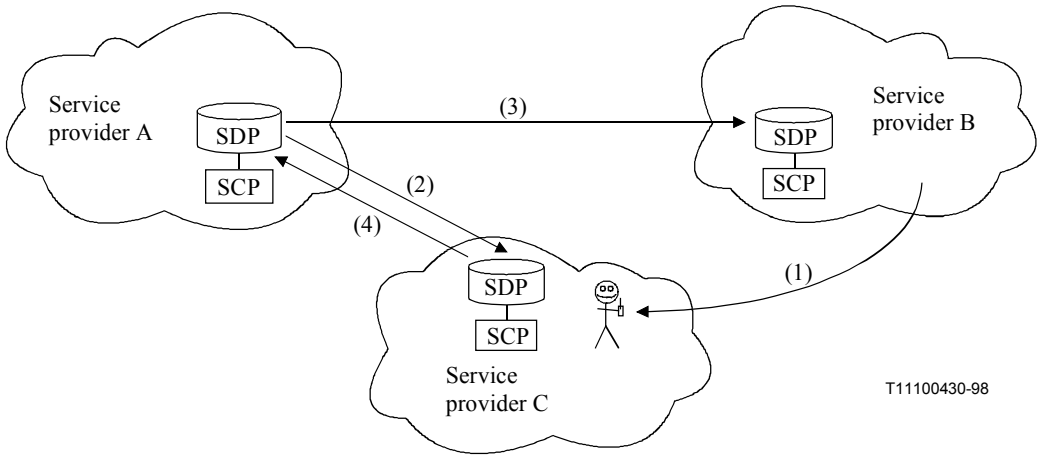


Figure A.12/Q.1229 – Subscriber roams from one visited service provider to another

When the subscriber roams to service provider C, service provider C detects the presence of the roaming user. The service provider then collects service and authentication information from the subscriber's equipment. From this information service provider C can determine the subscriber's unique identity and home service provider.

SCP C then modifies the visiting subscriber's profile, on the subscriber's home SDP, to indicate it as belonging to shadowing agreement AC. SDP A detects the change in its master information, deletes the copy of the subscriber's profile on SDP B, and then sends a copy of the subscriber's profile to SDP C.

Figure A.13 shows the case where directory operations are all sent in the same PDU and does not include message sequences for errors or for unbinding.

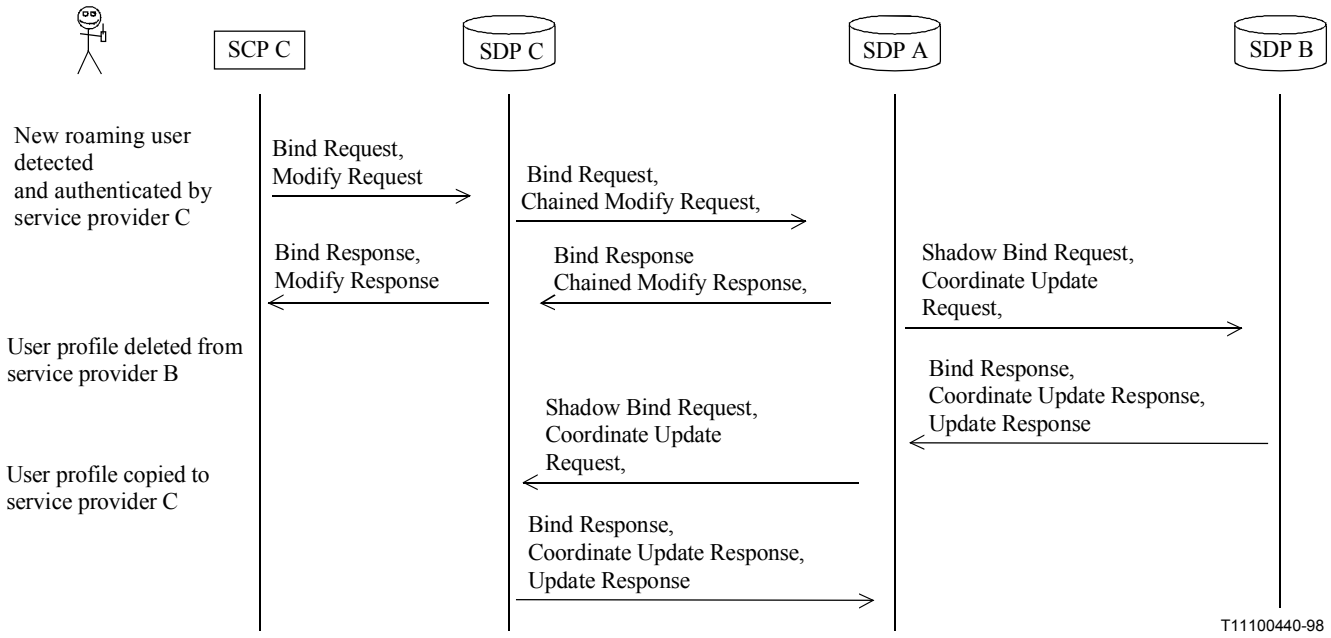


Figure A.13/Q.1229 – Visited service provider to visited service provider message sequence chart

A.5.2.4 Subscriber roams visited to home service provider

Suppose, finally, that the subscriber roams back to the home service provider, as shown in Figure A.14.

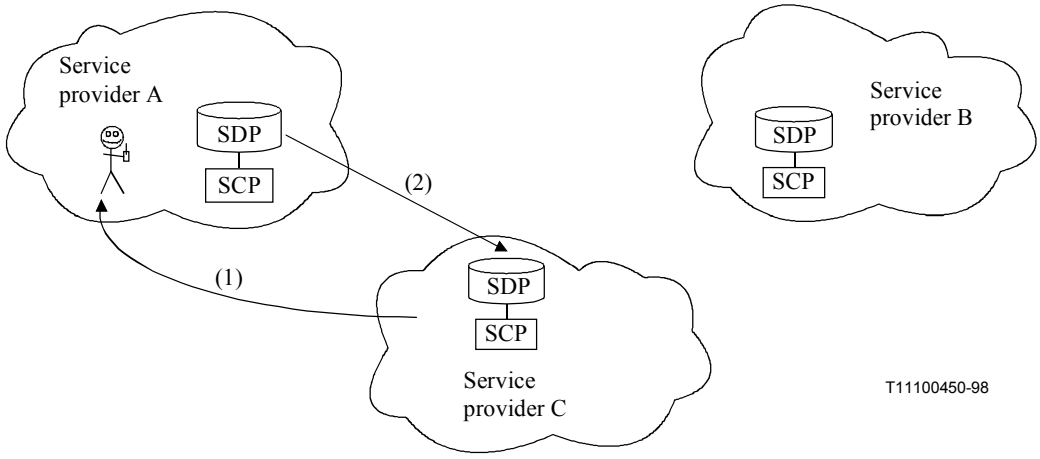


Figure A.14/Q.1229 – Subscriber roams from visited to home service provider

Service provider A detects the presence of the roaming user and determines the user's unique identity and home service provider.

SCP A then modifies the subscriber's profile to indicate it as not belonging to a shadowing agreement. SDP A subsequently detects the change in its master information and deletes the copy of the subscriber's profile on SDP C.

Figure A.15 shows the case where directory operations are all sent in the same PDU and does not include message sequences for errors or for unbinding.

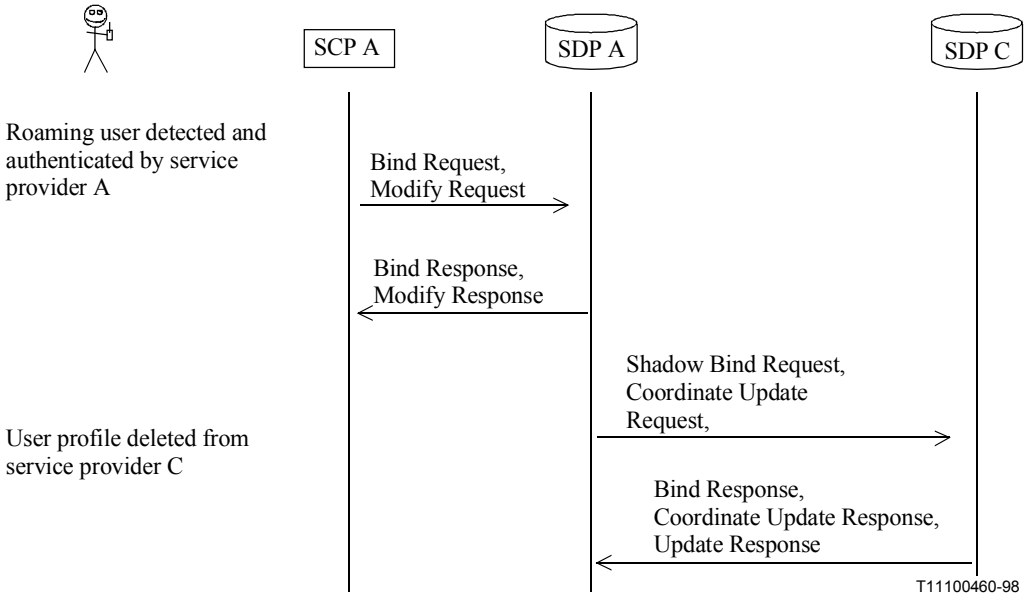


Figure A.15/Q.1229 – Visited to home message sequence chart

A.5.3 Assumptions

The assumptions made for this example scenario are summarized as follows:

1) A service provider must have some means of detecting the presence of a new roaming user.

2) A visited service provider must have some way to determine a roaming user's unique identity and home service provider.

3) The data structure (DIT) is understood between cooperating SDP pairs such that a distinguished name can be derived from knowledge of a user's unique identity and home service provider.

4) A roaming user's profile contains information which indicates whether it should be shadowed according to a particular shadowing agreement, if applicable.

A.5.4 Object modelling

This subclause describes the object modeling that could be used for Inter-Network Service Profile Transfer (ISPT).

A.5.4.1 Assumptions

A.5.4.1.1 Disclosure of profile entries

It is not desirable to require that the entire list of subscriber entries for a service provider be made available to the other service providers. Only the profiles of roaming users should be disclosed.

A.5.4.2 DIT schema

A.5.4.2.1 X.500 DIT

The X.500 DIT shown in Figure A.16 illustrates a sample directory tree. The tree is structured according to the components of the global E.164 numbering plan. In Figure A.16, the entries under the node **iNDigit = 1** represent components of North American Numbering Plan telephone numbers. Other subtrees may be structured according to other additional numbering plans, as required. It would be desirable to place the IN services tree in a position in the global directory that reflects its intention for international use. For this example, it is located under the following node in the tree:

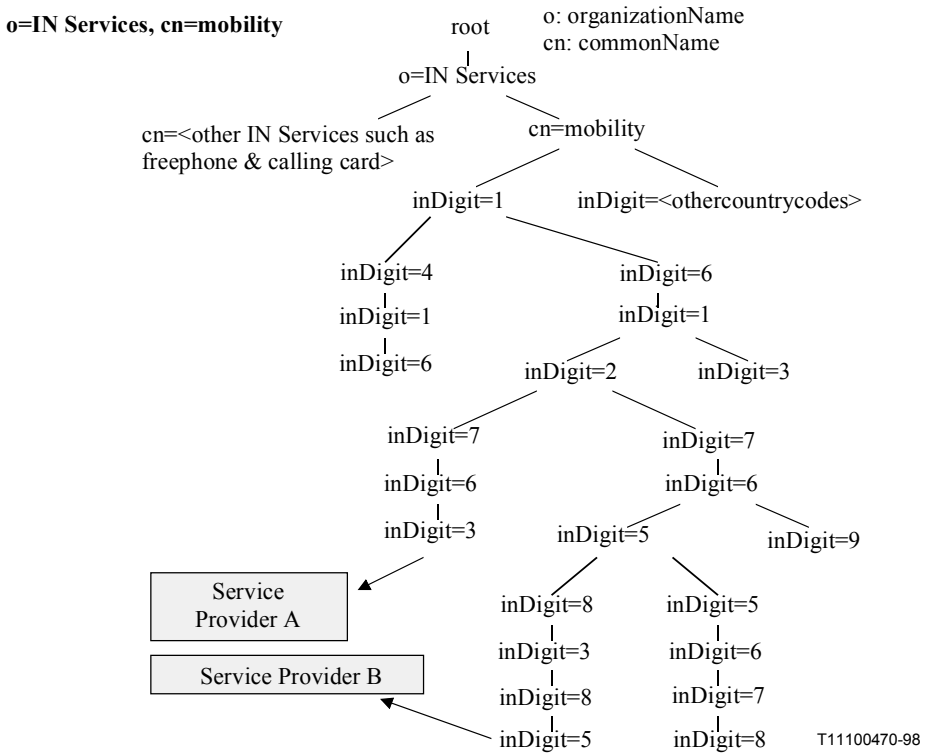


Figure A.16/Q.1229 – IN Services Global Directory

An SCP can, given a telephone number, construct the distinguished name (DN) of an entry in the tree so that it can read the entry's attributes without requiring a search. The sample structure can be used to encompass all possible numbering plans. Leaf nodes in the tree may contain aliases which point to private directories. Use of single digits minimizes the numbering plan knowledge required by an SCP and avoids a search operation.

It allows blocks of numbers to be easily divided between service providers.

The directory would be distributed, and possibly shared where necessary, between those interested in cooperating to provide such a service.

A.5.4.3 Access control

Aliases may be used to hide a corporate DSA's tree structure. If a portion of the tree is shadowed to a private directory, and the proper access controls are set, the private directory administrator can add leaf entries to the shadowed data. These leaf entries could then be shadowed back to the shared directory. The private directory, not the shared directory, would hold the leaf entry's directory name. For dereferenced aliases, the **denyReturnDN** permission should be used to prevent disclosure of the target X.500 DN to the query originator. Only an alias name should be returned in the search result.

To discourage repeated read operations from unauthorized users (trolling the directory), use of protected passwords or stronger authentication may be required. To prohibit access to entries without explicitly providing the name of an entry, **denyBrowse** should be enforced for all anonymous users.

A.5.4.4 Reducing message flow

When a user roams to another service provider, the visited service provider must modify the roaming user's profile on the user's home service provider directory. This will cause the user's profile to be transferred to the visited service provider's directory, as described the previous subclause.

In order to modify the user's profile, the visited service provider must determine the roaming user's directory name. A roaming user's telephone number can be used to derive the user's directory name in the tree.

A.5.4.5 Object classes

A.5.4.5.1 inMobilityUserProfile

The inMobilityUserProfile object class has been defined for storing profile information. The following ASN.1 definition may be used a starting point for describing the inMobilityUserProfile object class:

```
inMobilityUserProfile     OBJECT-CLASS  ::=     {
                KIND                      auxiliary
                SUBCLASS OF               {top}
                MUST CONTAIN              {inMobilityPIN |<other mandatory attributes}
                MAY CONTAIN    {          <optional attributes>}
                ID                        Id-oc-inMobilityUserProfile}
```

A.5.4.5.2 inNode

The inNode object class has been defined for defining entries in the tree. The following ASN.1 definition may be used a starting point for describing the inNode object class:

```
inNode          OBJECT-CLASS::=     {
                SUBCLASS OF         {top}
                MUST CONTAIN        {inDigit}
                ID                  Id-oc-inNode}
```

The inDigit attribute is the distinguished attribute.

A.5.4.6 Attribute types

A.5.4.6.1 inDigit

This attribute is used name entries in the IN services tree.

The ASN.1 definition for inDigit is as follows:

```
inDigit         ATTRIBUTE::= {
                WITH SYNTAX                     Digits (SIZE(1))
                EQUALITY MATCHING RULE          numericStringMatch
                ID                              id-at-inDigit}
```

A.5.4.6.2 inMobilityPIN

This attribute is used to store a mobility user's PIN number.

```
inMobilityPIN      ATTRIBUTE::= {
                   WITH SYNTAX userPassword (SIZE lbinMobilityPIN..ubinMobilityPIN)
                   ID                id-at-inMobilityPIN}
```

A.5.4.7 DIT structure definition

A.5.4.7.1 Name forms

A name form specifies the attribute that is to be used as the RDN for a specified object class.

A.5.4.7.2 inMobilityUserProfileNameForm

The following name form definition states that inMobilityID is the permitted distinguished attribute for the object class inMobilityUserProfile.

```
InNodeNameForm     NAME-FORM::= {
                   NAMES                        inMobilityUserProfile
                   WITH ATTRIBUTES              inDigit
                   ID                           id-nf-inNodeNameForm}
```

A.5.4.7.3 Structure rules

Structure rules specify permitted subordinate and superior entries in a DIT. The following structure rules which are illustrated in Figure A.17, can be used as a basis for defining the structure rules required for mobility service:

```
sr1        STRUCTURE-RULE::= {
                   NAME-FORM          countryNameForm
                   ID                 1}
sr2        STRUCTURE-RULE::= {
                   NAME-FORM          orgNameForm
                   SUPERIOR RULES     sr1
                   ID                 2}
sr2        STRUCTURE-RULE::= {
                   NAME-FORM          personNameForm
                   SUPERIOR RULES     sr2
                   ID                 3}
sr2        STRUCTURE-RULE::= {
                   NAME-FORM          inNodeNameForm
                   SUPERIOR RULES     sr3
                   ID                 4}
sr2        STRUCTURE-RULE::= {
                   NAME-FORM          inNodeNameForm
                   SUPERIOR RULES     sr4
                   ID                 5}
```

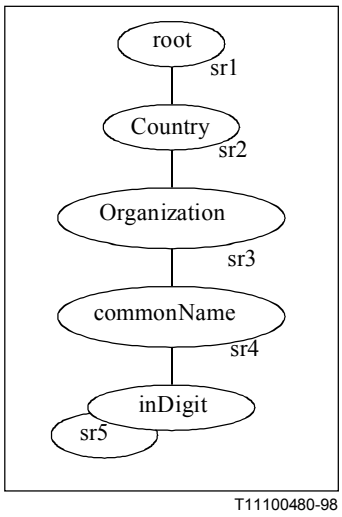


Figure A.17/Q.1229 – Structure rules

A.5.4.7.4 Object identifier assignments

The following object identifier assignments can be used as a starting point for identifying Mobility objects in Recommendation X.500.

```
id-at-inDigit           OBJECT IDENTIFIER::= {id-at-inMobility 0}
id-at-inMobilityPIN     OBJECT IDENTIFIER::= {id-at-inMobility 1}
id-oc-inNode            OBJECT IDENTIFIER::= {id-oc-inMobility 0}
id-nf-inNodeNameForm    OBJECT IDENTIFIER::= {id-nf-inMobility 0}
```

APPENDIX I

Service scenario examples for "Timed Disconnect" service features

I.1 Timed disconnect with announcement

Timed disconnect with announcement is a feature allowing the user to receive a tone or announcement that he/she will be disconnected after a certain period, and is subsequently disconnected after that period.

In the service example scenarios given, the SSF will initiate the start of a timer and play an announcement or a tone. Upon expiry of the timer, user interaction is ended and the SSF will release the call. The service scenarios uses the capability of performing user interaction in monitoring state (see Note).

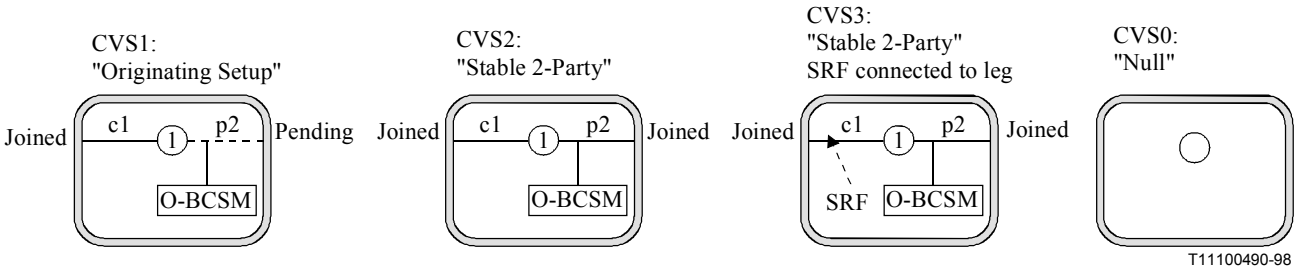
The featureTimed Disconnect with tone/announcement playing to the user may, e.g. be used together with the Call Disposition service feature, which provides the means to verify that the holders of a telecommunication card has enough spare credit (e.g. the card usage value has not been exceeded) to give permission to make the call. It implies that the usage of the card against the credit limit is tracked. This allows the support of services like Debit Cards, where charging may be controlled by the SCF identifying that the user has only few minutes left on his card.

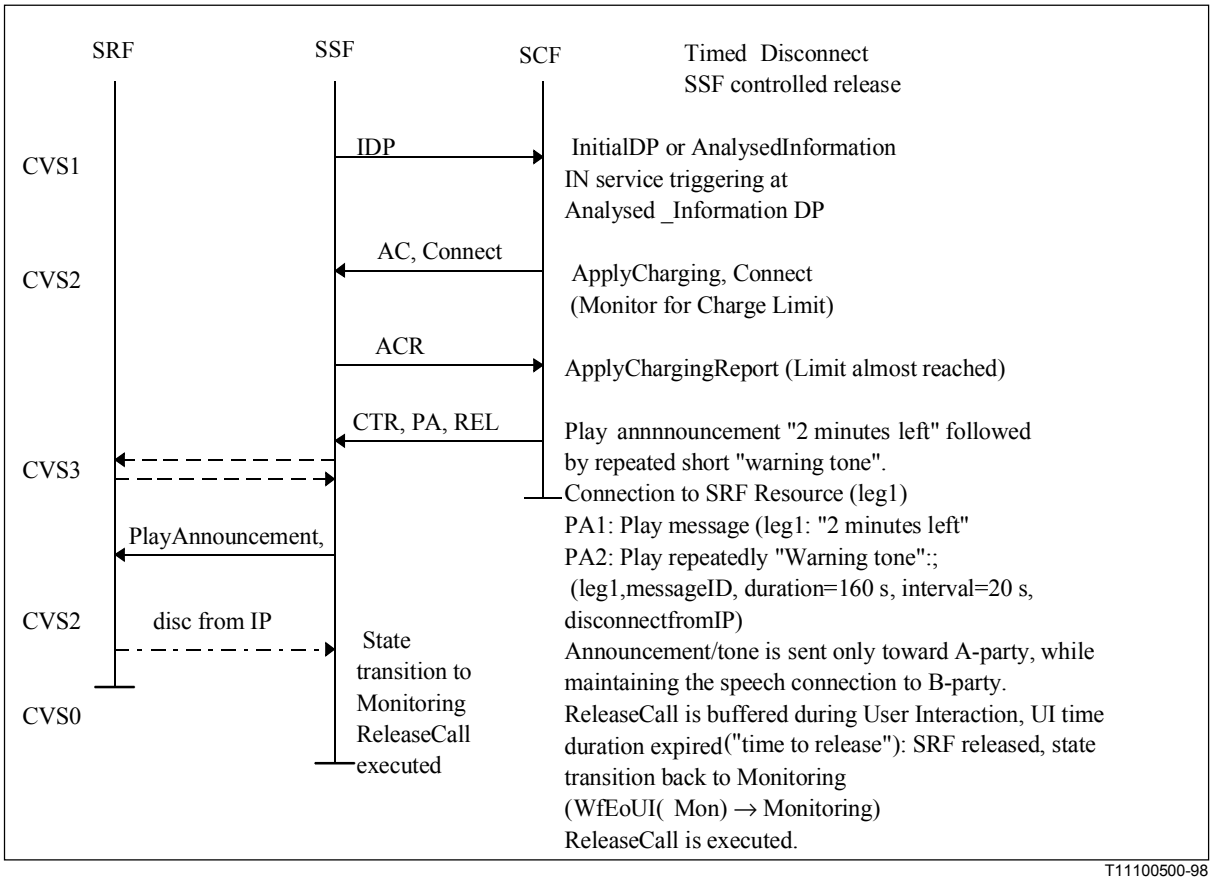
NOTE – The user interaction capability in monitoring state as shown in the service scenario examples will be in detail specified in IN CS-3 time frame.

I.2 Timed Disconnect with tone or announcement sending, SSF controlled release

In this example the SCF requests a Timed Disconnect with announcement or tone playing, whereby the SCF-SSF relationship is terminated. The SSF initiates announcement or tone sending to the user for a certain time period (i.e. time to release) and upon timer expiry SSF releases the call, i.e. SSF controlled release.

In this service scenario example a short warning tone/announcement is played repeatedly for a certain period (e.g. 2 minutes). When the announcement is ended after this certain period ("time to release") the ReleaseCall operation, which was buffered in the SSF during the user interaction, is executed and the call is released.



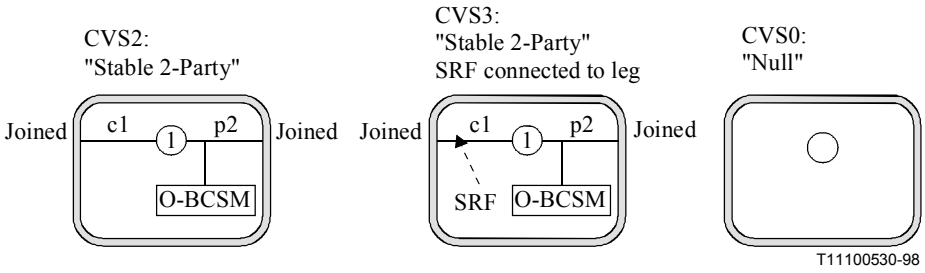


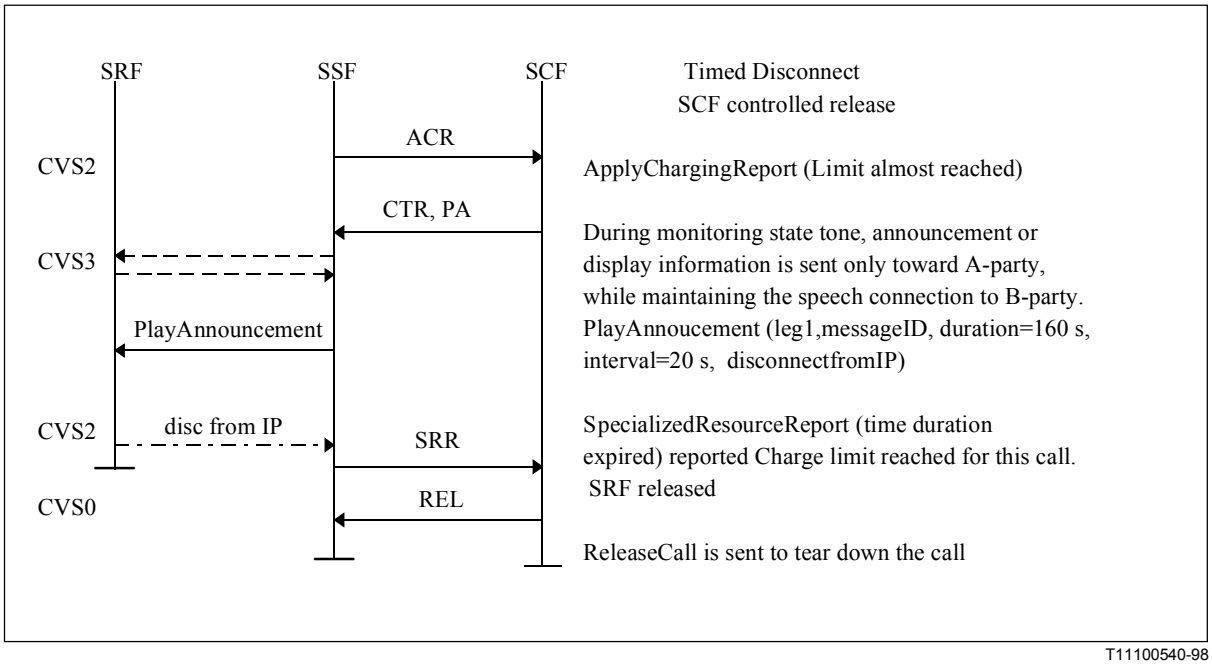
I.3 Timed Disconnect with tone or announcement sending, SCF controlled release

In this example a variant of the first service scenario is given; in this case a persistent control relationship between SCF and SSF is maintained until call release. The SCF is to receive a notification of timer expiry ("time to release") in order to request the call to released, i.e. SCF controlled release. This alternative would allow the user to avoid that the call be cleared by entering a password, or by entering another prepaid card number.

This information could for example be passed by the user via OCCRUI or due to "Mid-Call" event (EDP-R) processing allowing the user to interrupt call processing and notify the SCF of this event . The SCF could then via user interaction (e.g. announcement sending and DTMF reception) collect the necessary information from the user.

In this service scenario example a short warning tone/announcement is played repeatedly for a specified period (e.g. 2 minutes). When the announcement is ended a SpecializedResourceReport shall be sent to the SCF and the SCF may respond with ReleaseCall. This would also allow the user to avoid that the call be cleared by entering a password (e.g. MidCall events and USI can be reported).





APPENDIX II

Detailed SCCP Called and Calling address information

Applying the constraints from Annex B/Q.713 to the addresses in 7.2.3.9.3.3 gives the following formats for the elements in Table 7-25:

IF 2.1	adrS(**inap[FE-F1**])	=	adrS(ssn(**X**), gt(**FE-F**))	[Notes 1, 4, 6]
IF 2.2	adrP(**inap[FE-F2**])	=	adrP(rgt, ssn(**X**), gt(**FE-F**))	[Notes 1, 4, 6]
IF 2.3	adrP(**inap[FE-F3**])	=	adrP(rgt, ssn(**0**), gt(**FE-F**))	[Notes 1, 3, 6]
IF 2.4,	adrP(**inap[FE-F4**])	=	adrP(rpc, ssn(**Y**), gt(**FE-F**))	[Notes 1, 2, 5]
IF 2.5	adrS(**inap[FE-F4**])	=	adrS(ssn(**Y**), gt(**FE-F**))	[Notes 1, 2, 5]
IF 5.1	adrS(**inap[FE-F**])	=	adrS(rgt, ssn(**0**), gt(**FE-F**))	[Notes 1, 3, 6]
IF 2.1, 2.5, 5.1	adrS(**inap[FE-A**])	=	adrS(ssn(**0**), gt(**FE-A**))	[Notes 1, 3, 6]
IF 2.2-4	adrP(**inap[FE-A**])	=	adrP(rgt, ssn(**0**), gt(**FE-A**))	[Notes 1, 3, 6]
IF 5.2	adrP(**inap[FE-A2**])	=	adrP(rgt, ssn(**Y**), gt(**FE-A**))	[Notes 1, 5, 6]
IF 5.3	adrP(**inap[FE-A3**])	=	adrP(rgt, ssn(**0**), gt(**FE-A**))	[Notes 1, 3, 6]
IF 5.4	adrP(**inap[FE-A4**])	=	adrP(rpc, ssn(**X**), gt(**FE-A**))	[Notes 1, 2, 4]
IF 5.5	adrS(**inap[FE-A4**])	=	adrS(ssn(**X**), gt(**FE-A**))	[Notes 1, 2, 4]

NOTE 1 – The format of the global title field in each address is currently not standardized for INAP. Current internationally standardized formats for the GT field which are considered suitable are either that specified in B.4.3/Q.713[5] or that specified in B.4.4/Q.713[6].

It is advised that SCCP protocol and network design experts be involved in decisions related to SCCP message addressing. In the case of international internetworking this is WP 5/2 SG 11 ITU-T (Q.16).

NOTE 2 – The need for the global title field to be present in the final N-UNITDATA primitive passed to TC depends on the method chosen for INAP FE determination in 7.2.3.9.2.2. If the application context method is used, then the GT need not be present. Otherwise the GT must be present and must be the same as the GT in the original called party address in order for the destination FE to be determined. This would require that the GT value must be preserved during the GTT process.

NOTE 3 – The choice of SSN value 0 is mandatory for international use in the absence of a standardized SSN for INAP services.

NOTE 4 – The choice of SSN value **X** is a network specific matter within SCCP Network 1.

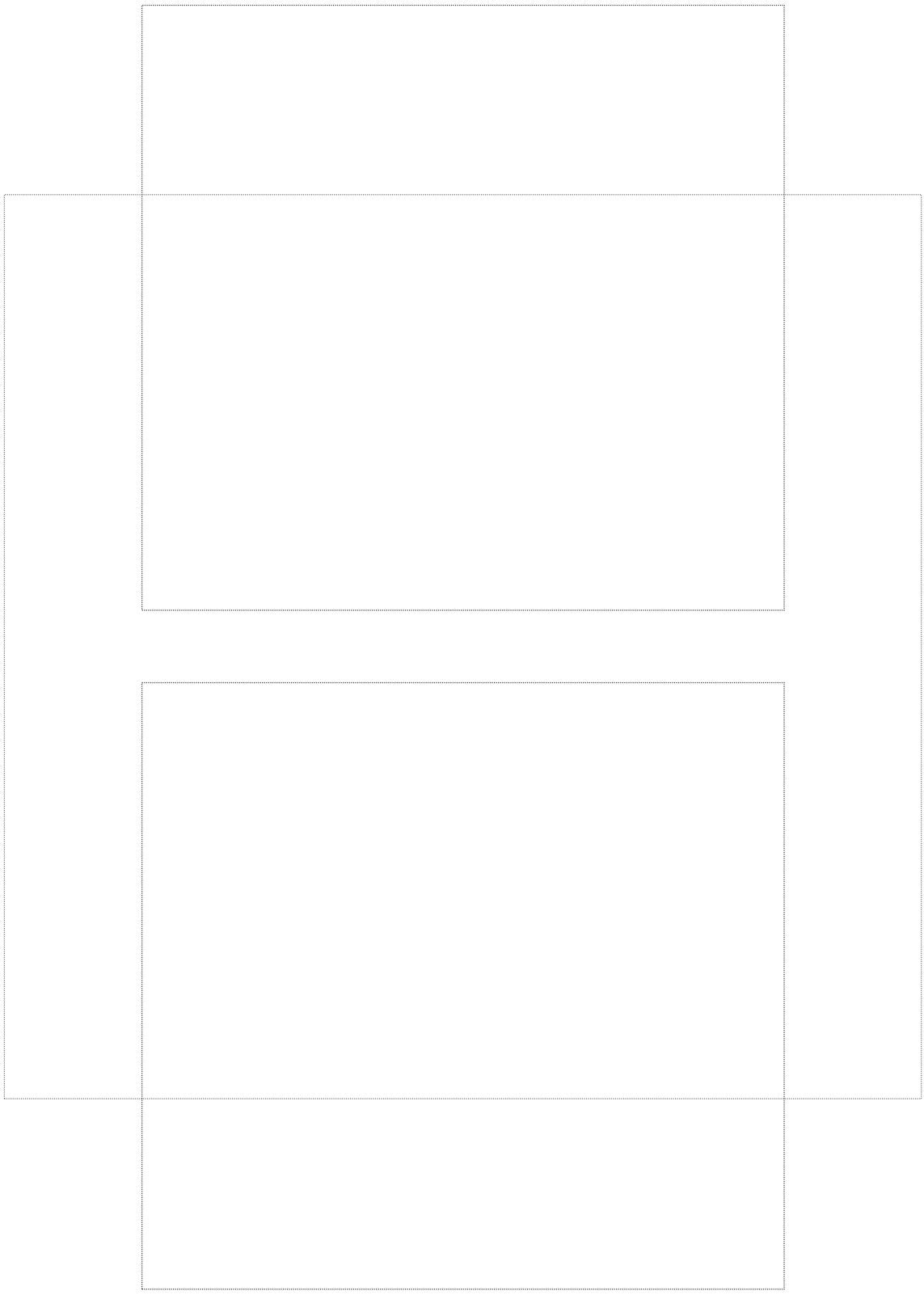
NOTE 5 – The choice of SSN value **Y** is a network specific matter within SCCP Network 2.

NOTE 6 – The value of the SSN at this point is arbitrary as it can be changed as a result of the next GTT process.

---

[5] This requires the GT to contain a Generic Number with a BCD Q.708 Z-UUU-V prefix.

[6] This requires the GT to contain an international E.164 number.



ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	**Switching and signalling**
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2000