



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**Q.1228**

**Fascículo 5/5**  
(09/97)

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN  
Red inteligente

---

**Recomendación sobre interfaces para el  
conjunto de capacidades 2 de red inteligente:  
Partes 5, 6 y 7**

Recomendación UIT-T Q.1228 – Fascículo 5/5

(Anteriormente Recomendación del CCITT)

---

RECOMENDACIONES DE LA SERIE Q DEL UIT-T  
**CONMUTACIÓN Y SEÑALIZACIÓN**

SEÑALIZACIÓN EN EL SERVICIO MANUAL INTERNACIONAL	Q.1–Q.3
EXPLORACIÓN INTERNACIONAL SEMIAUTOMÁTICA Y AUTOMÁTICA	Q.4–Q.59
FUNCIONES Y FLUJOS DE INFORMACIÓN PARA SERVICIOS DE LA RDSI	Q.60–Q.99
CLÁUSULAS APLICABLES A TODOS LOS SISTEMAS NORMALIZADOS DEL UIT-T	Q.100–Q.119
ESPECIFICACIONES DE LOS SISTEMAS DE SEÑALIZACIÓN N.º 4 Y N.º 5	Q.120–Q.249
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 6	Q.250–Q.309
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R1	Q.310–Q.399
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R2	Q.400–Q.499
CENTRALES DIGITALES	Q.500–Q.599
INTERFUNCIONAMIENTO DE LOS SISTEMAS DE SEÑALIZACIÓN	Q.600–Q.699
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 7	Q.700–Q.849
SISTEMA DE SEÑALIZACIÓN DIGITAL DE ABONADO N.º 1	Q.850–Q.999
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
<b>RED INTELIGENTE</b>	<b>Q.1200–Q.1999</b>
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999

*Para más información, véase la Lista de Recomendaciones del UIT-T.*

**Recomendación Q.1228**

**RECOMENDACIÓN SOBRE INTERFACES PARA EL CONJUNTO  
DE CAPACIDADES 2 DE RED INTELIGENTE**

**FASCÍCULO 5**

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.<sup>o</sup> 1 de la CMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

## NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

## PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 1999

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

## ÍNDICE

	Página
<b>PARTE 1</b>	
1 Introducción.....	1
2 Consideraciones generales.....	1
2.1 Referencias normativas .....	1
2.2 Abreviaturas y acrónimos.....	3
2.3 Convenios.....	9
3 Recomendación sobre los interfaces para servicios de telecomunicación .....	9
3.1 Consideraciones generales.....	9
3.1.1 Metodología de la definición .....	9
3.1.2 Ejemplo de escenarios físicos .....	9
3.1.3 Arquitectura del protocolo INAP .....	18
3.1.4 Direccionamiento INAP.....	19
3.1.5 Relación entre la Recomendación Q.1224 y esta Recomendación .....	20
3.1.6 Mecanismos de compatibilidad utilizados para el INAP .....	25
3.2 Reglas SACF/MACF.....	26
3.2.1 Reflexión del contexto de aplicación de la TCAP .....	26
3.2.2 Ejecución secuencial/paralela de operaciones.....	27
4 Tipos comunes del CS2 de red inteligente .....	27
4.1 Tipo de datos .....	27
4.2 Tipos de error .....	54
4.3 Códigos de operaciones.....	57
4.4 Códigos de error .....	60
4.5 Clases .....	61
4.6 Identificadores de objetos.....	69
5 Interfaz SSF/SCF.....	74
5.1 Operaciones y argumentos.....	74
5.2 Lotes, contratos y contextos de aplicación SSF/SCF .....	120
5.2.1 Visión general del protocolo .....	120
5.2.2 Módulo ASN.1 de SSF/SCF .....	139
6 Interfaz SCF/SRF .....	160
6.1 Operaciones y argumentos de SCF/SRF .....	160
6.2 Contratos, lotes y contextos de aplicación de SRF-SCF .....	165

	Página	
6.2.1	Visión general del protocolo .....	165
6.2.2	Módulos ASN.1 de SRF/SCF .....	166
7	Interfaz SCF-SDF.....	169
7.1	Introducción a la reutilización de la Recomendación X.500 para las interfaces SDF	169
7.1.1	Armonización entre los conceptos de la serie X.500 y la red inteligente .....	169
7.1.2	Utilización de un subconjunto limitado de la serie X.500 .....	170
7.1.3	Hipótesis de trabajo.....	170
7.2	El modelo de información de la SDF .....	170
7.2.1	El marco de referencia de la información .....	171
7.2.2	Control de acceso básico.....	172
7.2.3	Contextos de atributo .....	174
7.2.4	Definiciones de atributo .....	175
7.3	El protocolo de la interfaz SCF-SDF .....	176
7.3.1	Tipos de información y procedimientos comunes.....	176
7.3.2	Operaciones.....	178
7.3.3	Errores .....	181
7.4	Visión general del protocolo .....	182
7.4.1	Operaciones a distancia.....	182
7.4.2	Objetos y contratos ROS del directorio.....	182
7.4.3	Contratos y lotes del protocolo de acceso al directorio (DAP).....	183
7.5	Sintaxis abstracta del protocolo de directorio .....	184
7.5.1	Sintaxis abstractas .....	184
7.5.2	Contextos de aplicación del directorio.....	186
7.5.3	Códigos de operación.....	187
7.5.4	Códigos de error.....	187
7.5.5	Versiones y reglas de extensibilidad .....	187
7.6	Conformidad.....	189
7.6.1	Conformidad de las SCF .....	189
7.6.2	Conformidad de las SDF .....	189
7.7	Módulos ASN.1 para la interfaz SCF-SDF.....	191
7.7.1	Módulo del marco de información de la SDF del CS-2 de la red inteligente	191
7.7.2	Módulo de control de acceso básico de la SDF del CS-2 la red inteligente .	192
7.7.3	Módulo de las operaciones SCF-SDF del CS-2 la red inteligente .....	195
7.7.4	Módulo del protocolo SCF-SDF del CS-2 de red inteligente .....	196
8	Interfaz SDF/SDF.....	199
8.1	Introducción a los subconjuntos DISP y DSP de la Recomendación X.500 de red inteligente .....	199

	Página	
8.2	Hipótesis de trabajo.....	200
8.3	El subconjunto DISP de la serie X.500 de la red inteligente.....	200
8.3.1	Especificación del acuerdo de ensombrecimiento .....	200
8.3.2	Vinculación de la sombra del DSA.....	201
8.3.3	Desvinculación de la sombra del DSA de la red inteligente.....	201
8.3.4	Coordinación de la actualización de la sombra.....	201
8.3.5	Actualización de la sombra .....	202
8.3.6	Petición de actualización de sombra .....	203
8.4	El subconjunto del DSP de la Recomendación X.500 para la red inteligente.....	204
8.4.1	Tipos de información y procedimientos comunes.....	204
8.4.2	Vinculación del DSA .....	209
8.4.3	Desvinculación del DSA de red inteligente .....	209
8.4.4	Operaciones encadenadas.....	209
8.4.5	Errores encadenados.....	210
8.5	Visión general del protocolo .....	210
8.5.1	Objetos de operaciones a distancia y contratos.....	210
8.5.2	Contrato y lotes del protocolo del sistema de directorio.....	211
8.5.3	Contrato y lotes del DISP.....	212
8.6	Sintaxis abstracta del protocolo.....	213
8.6.1	Sintaxis abstracta del protocolo del sistema de directorio (DSP) .....	213
8.6.2	Sintaxis Abstracta del protocolo de ensombrecimiento de la información de directorio (DISP).....	214
8.6.3	Contexto de aplicación del sistema de directorio.....	214
8.6.4	Contexto de aplicación de sombra de directorio.....	215
8.6.5	Versiones y reglas de la extensibilidad .....	216
8.7	Conformidad.....	217
8.7.1	Conformidad de las SDF .....	218
8.7.2	Conformidad de un suministrador de sombra .....	219
8.7.3	Conformidad del consumidor de sombra .....	220
8.8	Módulos ASN.1 para la interfaz SDF-SDF.....	221
8.8.1	Módulo del protocolo SDF-SDF del CS-2 de red inteligente .....	221
9	El interfaz SCF/SCF.....	225
9.1	Argumentos y operaciones SCF/SCF.....	225
9.2	Contratos, lotes y contextos de aplicación de SCF/SCF .....	237
9.2.1	Visión general del protocolo .....	237
9.2.2	Módulos ASN.1 .....	242

	Página
10 Interfaz SCF/CUSF .....	247
10.1 Operaciones y argumentos.....	247
10.2 Contratos, lotes de operación y contextos de aplicación de la SCF/CUSF.....	252
10.2.1 Visión general del protocolo .....	252
10.2.2 Módulo ASN.1 .....	254
<b>PARTE 2</b>	
11 Procedimientos de entidad de aplicación SSF.....	259
11.1 Consideraciones generales.....	259
11.2 Modelo e interfaces .....	259
11.3 Relaciones entre el SSF FSM y la CCF y funciones de mantenimiento .....	260
11.4 Modelo de estados finitos para gestión de SSF (SSME FSM).....	263
11.5 FSM del modelo de estado de conmutación (SSM) para SSF .....	265
11.5.1 Modelo de estados finitos para asociación de segmentos de llamada (CSA)	268
11.5.2 Modelo de estados finitos para segmento de llamada.....	272
11.6 FSM de SSF asistente.....	284
11.6.1 Estado aa: Reposo .....	284
11.6.2 Estado ab: Espera de instrucciones .....	285
11.6.3 Estado ac: Espera de fin de interacción de usuario .....	286
11.7 FSM de SSF transferida .....	287
11.7.1 Estado ha: Reposo .....	287
11.7.2 Estado hb: Espera de instrucciones .....	288
11.7.3 Estado hc: Espera de fin de interacción de usuario.....	289
11.8 USI FSM de interacción servicio-usuario .....	290
12 Procedimientos de entidad de aplicación SCF .....	291
12.1 Consideraciones generales.....	291
12.2 Modelo e interfaces .....	292
12.3 Relación entre el SCF FSM y los SLP/funciones de mantenimiento .....	292
12.4 Diagrama parcial de transición de estados de la entidad de gestión de SCF (SCME)	295
12.4.1 Estado M1: Informe de estado en Reposo.....	296
12.4.2 Estado M2: Espera de respuesta de informe de estado de recurso de la SSF	296
12.4.3 Estado M3: Filtrado de servicio en Reposo .....	297
12.4.4 Estado M4: Espera de respuesta de filtrado de servicio de la SSF .....	297
12.4.5 Estado M5: Prueba de actividad en Reposo.....	297
12.4.6 Estado M6: Espera de respuesta de prueba de actividad.....	297
12.4.7 Estado M7: Gestión de datos de activador en reposo .....	298

	Página
12.4.8 Estado M8: Espera de respuesta de prueba de actividad de gestión de datos de activador .....	298
12.4.9 El objeto control de recursos.....	298
12.5 El modelo de estados de llamada SCF (SCSM).....	299
12.5.1 Estados afines a la SSF/SRF (SCSM-SSF/SRF).....	299
12.5.2 Estados afines a la SDF (SCSM-SDF).....	335
12.5.3 Estados afines a la SCF.....	336
12.5.4 Estados afines a la CUSF (SCSM-CUSF) .....	347
12.5.5 FSM de USI_SCF .....	350
13 Procedimientos de entidad de aplicación SRF .....	351
13.1 Consideraciones generales.....	351
13.2 Modelo e interfaces .....	352
13.3 Relación entre el SRF FSM y las funciones de mantenimiento o el tratamiento de la conexión soporte .....	353
13.4 El SRSM.....	354
13.4.1 Estado 1: Reposo.....	356
13.4.2 Estado 2: Conectado.....	357
13.4.3 Estado 3: Interacción de usuario .....	358
13.5 Ejemplo de procedimientos de control de SRF .....	360
13.5.1 Procedimientos de conexión de SRF .....	360
13.5.2 Procedimientos de interacción de usuario de extremo en SRF .....	364
13.5.3 Procedimientos de desconexión SRF .....	366
13.5.4 Ejemplos que ilustran secuencias completas de interacción de usuario.....	369
14 Procedimientos de entidad de aplicación SDF.....	376
14.1 Consideraciones generales.....	376
14.2 Modelo e interfaces .....	377
14.3 La estructura del SDF FSM.....	378
14.4 Modelos de transición de estado de SDF .....	379
14.4.1 Modelo de transición de estado de SDF para estados afines a SCF.....	379
14.4.2 Modelo de transición de estados de SDF para estados afines a SDF.....	381
15 Procedimientos de entidad de aplicación CUSF .....	398
15.1 Consideraciones generales.....	398
15.2 Modelo e interfaces .....	399
15.2.1 Fundamentos de la modelación y el protocolo.....	399
15.2.2 Modelación y protocolo .....	400
15.3 Relaciones entre el CUSF FSM y las SSF/CCF y funciones de mantenimiento.....	401
15.4 Modelo de estados finitos de la gestión de CUSF (CUSME FSM) .....	403

	Página
15.5	Diagrama de transición de estados CUSF ..... 403
15.5.1	Estado a: Reposo ..... 405
15.5.2	Estado b: Espera de instrucciones ..... 406
15.5.3	Estado c: Observación ..... 406
16	Procedimientos de error ..... 407
16.1	Procedimientos de error afines a operaciones ..... 407
16.1.1	Error de atributo ..... 407
16.1.2	Cancelado ..... 409
16.1.3	CancelFailed (cancelación fallida) ..... 410
16.1.4	DSAReferral ..... 411
16.1.5	ETCFailed ..... 412
16.1.6	ExecutionError ..... 412
16.1.7	ImproperCallerResponse ..... 414
16.1.8	MissingCustomerRecord ..... 416
16.1.9	MissingParameter (falta parámetro) ..... 424
16.1.10	Error de nombre ..... 435
16.1.11	ParameterOutOfRange ..... 437
16.1.12	Referral ..... 439
16.1.13	RequestedInfoError ..... 440
16.1.14	Error ScfReferral ..... 440
16.1.15	Seguridad ..... 443
16.1.16	Servicio ..... 451
16.1.17	Sombreado ..... 453
16.1.18	Fallo de sistema ..... 455
16.1.19	TaskRefused ..... 458
16.1.20	UnavailableResource ..... 460
16.1.21	UnexpectedComponentSequence ..... 462
16.1.22	UnexpectedDataValue ..... 465
16.1.23	UnexpectedParameter ..... 468
16.1.24	UnknownLegID ..... 471
16.1.25	Recurso desconocido ..... 471
16.1.26	Actualización ..... 472
16.1.27	ChainingRefused ..... 473
16.1.28	DirectoryBindError ..... 477
16.1.29	ScfBindFailure ..... 479
16.1.30	ScfTaskRefused ..... 480

	Página
16.2 Procedimientos de error afines a la entidad.....	481
16.2.1 Expiración de T <sub>SSF</sub> .....	481
16.2.2 Expiración de T <sub>SRF</sub> .....	482
16.2.3 Expiración de T <sub>cusf</sub> .....	482
<b>PARTE 3</b>	
17 Procedimientos de operación detallados .....	485
17.1 Procedimiento ActivateServiceFiltering .....	485
17.1.1 Descripción general.....	485
17.1.2 Entidad invocadora (SCF).....	488
17.1.3 Entidad respondedora (SSF) .....	489
17.2 Procedimiento ActivationReceivedAndAuthorized .....	490
17.2.1 Descripción general.....	490
17.2.2 Entidad invocadora (CUSF).....	491
17.2.3 Entidad respondedora (SCF) .....	491
17.3 Procedimiento ActivityTest.....	492
17.3.1 Descripción general.....	492
17.3.2 Entidad invocadora (SCF).....	492
17.3.3 Entidad respondedora (SSF) .....	492
17.3.4 Entidad respondedora (CUSF) .....	493
17.3.5 Entidad respondedora (SCF controladora o SCF soporte).....	493
17.4 Procedimiento AddEntry .....	494
17.4.1 Descripción general.....	494
17.4.2 Entidad invocadora (SCF).....	494
17.4.3 Entidad respondedora (SDF).....	494
17.5 Procedimiento AnalysedInformation.....	495
17.5.1 Descripción general.....	495
17.5.2 Entidad invocadora (SSF) .....	499
17.5.3 Entidad respondedora (SCF) .....	500
17.5.4 Tratamiento de errores .....	501
17.6 Procedimiento AnalyseInformation.....	501
17.6.1 Descripción general.....	501
17.6.2 Entidad invocadora (SCF) .....	502
17.6.3 Entidad respondedora (SSF) .....	503
17.7 Procedimiento ApplyCharging.....	505
17.7.1 Descripción general.....	505
17.7.2 Entidad invocadora (SCF) .....	506
17.7.3 Entidad respondedora (SSF) .....	506

	Página
17.8 Procedimiento ApplyChargingReport .....	507
17.8.1 Descripción general.....	507
17.8.2 Entidad invocadora (SSF) .....	507
17.8.3 Entidad respondedora (SCF) .....	508
17.9 Procedimiento AssistRequestInstructions .....	508
17.9.1 Descripción general.....	508
17.9.2 Entidad invocadora (SSF/SRF).....	509
17.9.3 Entidad respondedora (SCF) .....	509
17.10 Procedimiento AssociationReleaseRequested.....	510
17.10.1 Descripción general.....	510
17.10.2 Entidad invocadora (CUSF).....	511
17.10.3 Entidad respondedora (SCF) .....	511
17.11 Procedimiento AuthorizeTermination .....	512
17.11.1 Descripción general.....	512
17.11.2 Entidad invocadora (SSF/SRF).....	513
17.11.3 Entidad respondedora (SSF) .....	513
17.12 Procedimiento CallGap .....	514
17.12.1 Descripción general.....	514
17.12.2 Entidad invocadora (SCF).....	516
17.12.3 Entidad respondedora (SSF) .....	517
17.13 Procedimiento CallInformationReport .....	518
17.13.1 Descripción general.....	518
17.13.2 Entidad invocadora (SSF) .....	519
17.13.3 Entidad respondedora (SCF) .....	520
17.13.4 Tratamiento de errores .....	520
17.14 Procedimiento CallInformationRequest .....	521
17.14.1 Descripción general.....	521
17.14.2 Entidad invocadora (SCF).....	522
17.14.3 Entidad respondedora (SSF) .....	522
17.15 Procedimiento Cancel.....	523
17.15.1 Descripción general.....	523
17.15.2 Entidad invocadora (SCF).....	523
17.15.3 Entidad respondedora (SRF).....	524
17.15.4 Entidad respondedora (SSF) .....	524
17.16 Procedimiento CancelStatusReportRequest .....	524
17.16.1 Descripción general.....	524
17.16.2 Entidad invocadora (SCF).....	525
17.16.3 Entidad respondedora (SSF) .....	525

	Página
17.17 Procedimiento chainedAddEntry.....	525
17.17.1 Descripción general.....	525
17.17.2 Entidad invocadora (SDF).....	526
17.17.3 Entidad respondedora (SDF).....	526
17.18 Procedimiento chainedConfirmedNotificationProvided .....	527
17.18.1 Descripción general.....	527
17.18.2 Entidad invocadora (SCF soporte del iniciador de encadenamiento) .....	528
17.18.3 Entidad respondedora (SCF soporte del terminador de encaminamiento)....	528
17.19 Procedimiento ChainedConfirmedReportChargingInformation .....	529
17.19.1 Descripción general.....	529
17.19.2 Entidad invocadora (SCF soporte de iniciador de encadenamiento) .....	529
17.19.3 Entidad respondedora (SCF soporte de terminador de encadenamiento) ....	529
17.20 Procedimiento ChainedEstablishChargingRecord .....	530
17.20.1 Descripción general.....	530
17.20.2 Entidad invocadora (SCF soporte del terminador de encaminamiento) .....	530
17.20.3 Entidad respondedora (SCF soporte del terminador de encaminamiento)....	531
17.21 Procedimiento chainedExecute .....	531
17.21.1 Descripción general.....	531
17.21.2 Entidad invocadora (SDF).....	531
17.21.3 Entidad respondedora (SDF).....	532
17.22 Procedimiento ChainedHandlingInformationRequest.....	533
17.22.1 Descripción general.....	533
17.22.2 Entidad invocadora (SCF soporte del iniciador de encadenamiento) .....	534
17.22.3 Entidad respondedora (SCF soporte del terminador de encadenamiento) ....	534
17.23 Procedimiento ChainedHandlingInformationResult .....	534
17.23.1 Descripción general.....	534
17.23.2 Entidad invocadora (SCF soporte del terminador de encaminamiento) .....	535
17.23.3 Entidad respondedora (SCF soporte del iniciador de encaminamiento).....	535
17.24 Procedimiento chainedModifyEntry.....	536
17.24.1 Descripción general.....	536
17.24.2 Entidad invocadora (SDF).....	536
17.24.3 Entidad respondedora (SDF).....	537
17.25 Procedimiento ChainedNetworkCapability.....	537
17.25.1 Descripción general.....	537
17.25.2 Entidad invocadora (SCF soporte del terminador de encaminamiento) .....	538
17.25.3 Entidad respondedora (SCF soporte del iniciador de encaminamiento).....	539

	Página
17.26 Procedimiento ChainedNotificationProvided.....	539
17.26.1 Descripción general.....	539
17.26.2 Entidad invocadora (SCF soporte del terminador de encaminamiento) .....	540
17.26.3 Entidad respondedora (SCF soporte del terminador de encaminamiento)....	540
17.27 Procedimiento ChainedReportChargingInformation.....	540
17.27.1 Descripción general.....	540
17.27.2 Entidad invocadora (SCF soporte del terminador de encaminamiento) .....	541
17.27.3 Entidad respondedora (SCF soporte del terminador de encaminamiento)....	542
17.28 Procedimiento ChainedProvideUserInformation .....	542
17.28.1 Descripción general.....	542
17.28.2 Entidad invocadora (SCF soporte del terminador de encaminamiento) .....	543
17.28.3 Entidad respondedora (SCF soporte del iniciador de encaminamiento).....	543
17.29 Procedimiento chainedRemoveEntry .....	543
17.29.1 Descripción general.....	543
17.29.2 Entidad invocadora (SDF).....	544
17.29.3 Entidad respondedora (SDF).....	544
17.30 Procedimiento ChainedRequestNotification .....	545
17.30.1 Descripción general.....	545
17.30.2 Entidad invocadora (SCF soporte del terminador de encaminamiento) .....	546
17.30.3 Entidad respondedora (SCF soporte del iniciador de encaminamiento).....	546
17.31 Procedimiento chainedSearch .....	546
17.31.1 Descripción general.....	546
17.31.2 Entidad invocadora (SDF).....	547
17.31.3 Entidad respondedora (SDF).....	548
17.32 Procedimiento CollectedInformation .....	548
17.32.1 Descripción general.....	548
17.32.2 Entidad invocadora (SSF) .....	550
17.32.3 Entidad respondedora (SCF).....	552
17.33 Procedimiento CollectInformation .....	553
17.33.1 Descripción general.....	553
17.33.2 Entidad invocadora (SCF).....	554
17.33.3 Entidad respondedora (SSF) .....	554
17.34 Procedimiento ComponentReceived .....	555
17.34.1 Descripción general.....	555
17.34.2 Entidad invocadora (CUSF).....	556
17.34.3 Entidad respondedora (SCF) .....	556

	Página
17.35 Procedimiento ConfirmedNotificationProvided.....	557
17.35.1 Descripción general.....	557
17.35.2 Entidad invocadora (SCF controladora).....	557
17.35.3 Entidad respondedora (SCF soporte) .....	558
17.36 Procedimiento ConfirmedReportChargingInformation.....	558
17.36.1 Descripción general.....	558
17.36.2 Entidad invocadora (SCF controladora).....	559
17.36.3 Entidad respondedora (SCF soporte) .....	560
17.37 Procedimiento Connect .....	560
17.37.1 Descripción general.....	560
17.37.2 Entidad invocadora (SCF).....	563
17.37.3 Entidad respondedora (SSF) .....	563
17.38 Procedimiento ConnectToResource .....	565
17.38.1 Descripción general.....	565
17.38.2 Entidad invocadora (SCF).....	565
17.38.3 Entidad respondedora (SSF) .....	566
17.39 Procedimiento Continue .....	566
17.39.1 Descripción general.....	566
17.39.2 Entidad invocadora (SCF).....	567
17.39.3 Entidad respondedora (SSF) .....	567
17.40 Procedimiento ContinueWithArgument.....	567
17.40.1 Descripción general.....	567
17.40.2 Entidad invocadora (SCF).....	568
17.40.3 Entidad respondedora (SSF) .....	569
17.41 Procedimiento CoordinateShadowUpdate .....	569
17.41.1 Descripción general.....	569
17.41.2 Entidad del suministrador (SDF) .....	570
17.41.3 Entidad del consumidor (SDF).....	571
17.42 Procedimiento CreateCallSegmentAssociation.....	572
17.42.1 Descripción general.....	572
17.42.2 Entidad invocadora (SCF).....	572
17.42.3 Entidad respondedora (SSF) .....	573
17.43 Procedimiento in-directoryBind .....	573
17.43.1 Descripción general.....	573
17.43.2 Entidad invocadora (SCF).....	573
17.43.3 Entidad respondedora (SDF).....	574

	Página
17.44 Procedimiento DirectoryUnbind .....	574
17.44.1 Descripción general.....	574
17.44.2 Entidad invocadora (SCF).....	574
17.44.3 Entidad respondedora (SDF).....	575
17.45 Procedimiento DisconnectForwardConnection.....	575
17.45.1 Descripción general.....	575
17.45.2 Entidad invocadora (SCF).....	575
17.45.3 Entidad respondedora (SSF) .....	576
17.46 Procedimiento DisconnectForwardConnectionWithArgument.....	577
17.46.1 Descripción general.....	577
17.46.2 Entidad invocadora (SCF).....	577
17.46.3 Entidad respondedora (SSF) .....	578
17.47 Procedimiento DisconnectLeg.....	578
17.47.1 Descripción general.....	578
17.47.2 Entidad invocadora (SCF).....	579
17.47.3 Entidad respondedora (SSF) .....	579
17.48 Procedimiento dSABind.....	580
17.48.1 Descripción general.....	580
17.48.2 Entidad invocadora (SDF).....	580
17.48.3 Entidad respondedora (SDF).....	580
17.49 Procedimiento de DSASHadowBind .....	581
17.49.1 Descripción general.....	581
17.49.2 Entidad suministradora (SDF).....	582
17.49.3 Entidad del consumidor (SDF).....	584
17.50 Procedimiento in-DSASHadowUnbind .....	585
17.50.1 Descripción general.....	585
17.50.2 Entidad del suministrador (SDF) .....	585
17.50.3 Entidad del consumidor (SDF).....	586
17.51 Procedimiento EntityReleased.....	587
17.51.1 Descripción general.....	587
17.51.2 Entidad invocadora (SSF) .....	588
17.51.3 Entidad respondedora (SCF) .....	588
17.52 Procedimiento EstablishChargingRecord.....	588
17.52.1 Descripción general.....	588
17.52.2 Entidad invocadora (SCF soporte).....	589
17.52.3 Entidad respondedora (SCF controladora).....	590

	Página
17.53 Procedimiento establishTemporaryConnection.....	590
17.53.1 Descripción general.....	590
17.53.2 Entidad invocadora (SCF).....	591
17.53.3 Entidad respondedora (SSF) .....	592
17.54 Procedimiento EventNotificationCharging .....	592
17.54.1 Descripción general.....	592
17.54.2 Entidad invocadora (SSF) .....	593
17.54.3 Entidad respondedora (SCF).....	594
17.55 Procedimiento EventReportBCSM .....	594
17.55.1 Descripción general.....	594
17.55.2 Entidad invocadora (SSF) .....	596
17.55.3 Entidad respondedora (SCF).....	596
17.56 Procedimiento EventReportFacility .....	597
17.56.1 Descripción general.....	597
17.56.2 Entidad invocadora (SSF) .....	598
17.56.3 Entidad respondedora (SCF).....	598
17.57 Procedimiento Execute.....	598
17.57.1 Descripción general.....	598
17.57.2 Entidad invocadora (SCF).....	599
17.57.3 Entidad respondedora (SDF).....	600
17.58 Procedimiento FacilitySelectedAndAvailable.....	601
17.58.1 Descripción general.....	601
17.58.2 Entidad invocadora (SSF) .....	602
17.58.3 Entidad respondedora (SCF).....	603
17.59 Procedimiento FurnishChargingInformation.....	603
17.59.1 Descripción general.....	603
17.59.2 Entidad invocadora (SCF).....	604
17.59.3 Entidad respondedora (SCF).....	605
17.60 Procedimiento HandlingInformationRequest.....	606
17.60.1 Descripción general.....	606
17.60.2 Entidad invocadora (SCF controladora).....	607
17.60.3 Entidad respondedora (SCF soporte) .....	608
17.61 Procedimiento HandlingInformationResult.....	609
17.61.1 Descripción general.....	609
17.61.2 Entidad invocadora (SCF soporte) .....	610
17.61.3 Entidad respondedora (SCF controladora).....	610

	Página
17.62 Procedimiento HoldCallInNetwork.....	611
17.62.1 Descripción general.....	611
17.62.2 Entidad invocadora (SCF).....	611
17.62.3 Entidad respondedora (SSF) .....	611
17.63 Procedimiento in-DSAUnbind .....	612
17.63.1 Descripción general.....	612
17.63.2 Entidad invocadora (SDF).....	612
17.63.3 Entidad respondedora (SDF).....	612
17.64 Procedimiento InitialDP .....	613
17.64.1 Descripción general.....	613
17.64.2 Entidad invocadora (SSF) .....	616
17.64.3 Entidad respondedora (SCF) .....	617
17.65 Procedimiento InitiateAssociation .....	618
17.65.1 Descripción general.....	618
17.65.2 Entidad invocadora (SCF).....	618
17.65.3 Entidad respondedora (CUSF) .....	618
17.66 Procedimiento InitiateCallAttempt.....	618
17.66.1 Descripción general.....	618
17.66.2 Entidad invocadora (SCF).....	620
17.66.3 Entidad respondedora (SSF) .....	620
17.67 Procedimiento ManageTriggerData .....	621
17.67.1 Descripción general.....	621
17.67.2 Entidad invocadora (SCF).....	622
17.67.3 Entidad respondedora (SSF) .....	622
17.68 Procedimiento MergeCallSegments .....	623
17.68.1 Descripción general.....	623
17.68.2 Entidad invocadora (SCF).....	623
17.68.3 Entidad respondedora (SSF) .....	623
17.69 Procedimiento ModifyEntry .....	624
17.69.1 Descripción general.....	624
17.69.2 Entidad invocadora (SCF).....	624
17.69.3 Entidad respondedora (SDF).....	625
17.70 Procedimiento MoveCallSegments .....	625
17.70.1 Descripción general.....	626
17.70.2 Entidad invocadora (SCF).....	627
17.70.3 Entidad respondedora (SSF) .....	627

	Página
17.71 Procedimiento MoveLeg .....	627
17.71.1 Descripción general.....	627
17.71.2 Entidad invocadora (SCF).....	628
17.71.3 Entidad respondedora (SSF) .....	628
17.72 Procedimiento NetworkCapability .....	629
17.72.1 Descripción general.....	629
17.72.2 Entidad invocadora (SCF soporte).....	629
17.72.3 Entidad respondedora (SCF controladora).....	630
17.73 Procedimiento NotificationProvided.....	630
17.73.1 Descripción general.....	630
17.73.2 Entidad invocadora (SCF controladora).....	631
17.73.3 Entidad respondedora (SCF soporte) .....	631
17.74 Procedimiento OAbandon .....	632
17.74.1 Descripción general.....	632
17.74.2 Entidad invocadora (SSF) .....	632
17.74.3 Entidad respondedora (SCF) .....	633
17.75 Procedimiento OAnswer .....	633
17.75.1 Descripción general.....	633
17.75.2 Entidad invocadora (SSF) .....	634
17.75.3 Entidad respondedora (SCF) .....	635
17.76 Procedimiento OCalledPartyBusy.....	636
17.76.1 Descripción general.....	636
17.76.2 Entidad invocadora (SSF) .....	637
17.76.3 Entidad respondedora (SCF) .....	637
17.77 Procedimiento ODisconnect.....	638
17.77.1 Descripción general.....	638
17.77.2 Entidad invocadora (SSF) .....	639
17.77.3 Entidad respondedora (SCF) .....	640
17.78 Procedimiento OMidCall .....	640
17.78.1 Descripción general.....	640
17.78.2 Entidad invocadora (SSF) .....	641
17.78.3 Entidad respondedora (SCF) .....	642
17.79 Procedimiento ONoAnswer .....	643
17.79.1 Descripción general.....	643
17.79.2 Entidad invocadora (SSF) .....	643
17.79.3 Entidad respondedora.....	644

	Página
17.80 Procedimiento OriginationAttempt .....	645
17.80.1 Descripción general.....	645
17.80.2 Entidad invocadora (SSF) .....	646
17.80.3 Entidad respondedora (SCF) .....	646
17.81 Procedimiento OriginationAttemptAuthorized .....	647
17.81.1 Descripción general.....	647
17.81.2 Entidad invocadora (SSF) .....	648
17.81.3 Entidad respondedora (SCF) .....	648
17.82 Procedimiento OSuspended .....	649
17.82.1 Descripción general.....	649
17.82.2 Entidad invocadora (SSF) .....	649
17.82.3 Entidad respondedora (SCF) .....	650
17.83 Procedimiento PlayAnnouncement .....	650
17.83.1 Descripción general.....	650
17.83.2 Entidad invocadora (SCF) .....	652
17.83.3 Entidad respondedora (SRF) .....	652
17.84 Procedimiento PromptAndCollectUserInformation.....	653
17.84.1 Descripción general.....	653
17.84.2 Entidad invocadora (SCF) .....	657
17.84.3 Entidad respondedora (SRF) .....	657
17.85 Procedimiento PromptAndReceiveMessage .....	659
17.85.1 Descripción general.....	659
17.85.2 Entidad invocadora (SCF) .....	661
17.85.3 Entidad respondedora (SRF) .....	662
17.86 Procedimiento ProvideUserInfo.....	663
17.86.1 Descripción general.....	663
17.86.2 Entidad invocadora (SCF soporte) .....	664
17.86.3 Entidad respondedora (SCF controladora) .....	664
17.87 Procedimiento Reconnect.....	665
17.87.1 Descripción general.....	665
17.87.2 Entidad invocadora (SCF) .....	666
17.87.3 Entidad respondedora (SSF) .....	666
17.88 Procedimiento ReleaseAssociation .....	667
17.88.1 Descripción general.....	667
17.88.2 Entidad invocadora (SCF) .....	667
17.88.3 Entidad respondedora (CUSF) .....	667

	Página
17.89 Procedimiento ReleaseCall.....	667
17.89.1 Descripción general.....	667
17.89.2 Entidad invocadora (SCF).....	668
17.89.3 Entidad respondedora (SSF) .....	669
17.90 Procedimiento RemoveEntry.....	669
17.90.1 Descripción general.....	669
17.90.2 Entidad invocadora (SCF).....	669
17.90.3 Entidad respondedora (SDF).....	670
17.91 Procedimiento ReportChargingInformation.....	670
17.91.1 Descripción general.....	670
17.91.2 Entidad invocadora (SCF controladora).....	671
17.91.3 Entidad respondedora (SCF soporte) .....	671
17.92 Procedimiento ReportUTSI.....	672
17.92.1 Descripción general.....	672
17.92.2 Entidad invocadora (SSF) .....	673
17.92.3 Entidad respondedora (SCF) .....	673
17.93 Procedimiento RequestCurrentStatusReport.....	673
17.93.1 Descripción general.....	673
17.93.2 Entidad invocadora (SCF).....	674
17.93.3 Entidad respondedora (SSF) .....	674
17.94 Procedimiento RequestEveryStatusChangeReport .....	674
17.94.1 Descripción general.....	674
17.94.2 Entidad invocadora (SCF).....	675
17.94.3 Entidad respondedora (SSF) .....	675
17.95 Procedimiento RequestFirstStatusMatchReport.....	676
17.95.1 Descripción general.....	676
17.95.2 Entidad invocadora (SCF).....	677
17.95.3 Entidad respondedora (SSF) .....	677
17.96 Procedimiento RequestNotification .....	678
17.96.1 Descripción general.....	678
17.96.2 Entidad invocadora (SCF soporte) .....	678
17.96.3 Entidad respondedora (SCF controladora).....	679
17.97 Procedimiento RequestNotificationChargingEvent .....	679
17.97.1 Descripción general.....	679
17.97.2 Entidad invocadora (SCF).....	680
17.97.3 Entidad respondedora (SSF) .....	680

	Página
17.98 Procedimiento RequestReportBCSMEvent .....	681
17.98.1 Descripción general.....	681
17.98.2 Entidad invocadora (SCF).....	683
17.98.3 Entidad respondedora (SSF) .....	683
17.99 Procedimiento RequestReportBCUSMEvent.....	684
17.99.1 Descripción general.....	684
17.99.2 Parámetros.....	684
17.99.3 Entidad invocadora (SCF).....	684
17.99.4 Entidad respondedora (CUSF) .....	685
17.100 Procedimiento RequestReportFacilityEvent .....	685
17.100.1 Descripcion general .....	685
17.100.2 Entidad invocadora (SCF).....	686
17.100.3 Entidad respondedora (SSF) .....	686
17.101 Procedimiento RequestReportUTSI.....	686
17.101.1 Descripción general .....	686
17.101.2 Entidad invocadora (SCF).....	687
17.101.3 Entidad respondedora (SSF) .....	688
17.102 Procedimiento RequestShadowUpdate .....	688
17.102.1 Descripción general .....	688
17.102.2 Entidad suministradora (SDF) .....	688
17.102.3 Entidad del consumidor (SDF) .....	689
17.103 Procedimiento ResetTimer .....	690
17.103.1 Descripción general .....	690
17.103.2 Entidad invocadora (SCF).....	690
17.103.3 Entidad respondedora (SSF) .....	691
17.104 Procedimiento RouteSelectFailure .....	691
17.104.1 Descripción general .....	691
17.104.2 Entidad invocadora (SSF).....	692
17.104.3 Entidad respondedora (SCF) .....	693
17.105 Procedimiento SCFBind.....	693
17.105.1 Descripción general .....	693
17.105.2 Entidad respondedora (SCF soporte) .....	694
17.106 Procedimiento scfBind (en el caso de encadenamiento) .....	695
17.106.1 Descripción general .....	695
17.106.2 Entidad invocadora (SCF soporte del iniciador de encaminamiento).....	696
17.106.3 Entidad respondedora (SCF soporte del terminador de encaminamiento)	696

	Página
17.107 Procedimiento SCFUnBind.....	696
17.107.1 Descripción general .....	696
17.107.2 Entidad invocadora (SCF controladora) .....	697
17.107.3 Entidad respondedora (SCF soporte) .....	697
17.108 Procedimiento scfUnBind (en el caso de encadenamiento) .....	697
17.108.1 Descripción general .....	697
17.108.2 Entidad invocadora (SCF soporte del terminador de encadenamiento)....	697
17.108.3 Entidad respondedora (SCF soporte del terminador de encadenamiento)	698
17.109 Procedimiento ScriptClose .....	698
17.109.1 Descripción general .....	698
17.109.2 Entidad invocadora (SCF).....	698
17.109.3 Entidad respondedora (SRF).....	699
17.110 Procedimiento ScriptEvent.....	699
17.110.1 Descripción general .....	699
17.110.2 Entidad invocadora (SRF).....	700
17.110.3 Entidad respondedora (SCF).....	700
17.111 Procedimiento ScriptInformation .....	701
17.111.1 Descripción general .....	701
17.111.2 Entidad invocadora (SCF).....	702
17.111.3 Entidad respondedora (SRF controladora).....	702
17.112 Procedimiento ScriptRun .....	703
17.112.1 Descripción general .....	703
17.112.2 Entidad invocadora (SCF).....	703
17.112.3 Entidad respondedora (SRF).....	704
17.113 Procedimiento Search.....	704
17.113.1 Descripción general .....	704
17.113.2 Entidad invocadora (SCF).....	704
17.113.3 Entidad respondedora (SDF).....	705
17.114 Procedimiento SelectFacility.....	706
17.114.1 Descripción general .....	706
17.114.2 Entidad invocadora (SCF).....	707
17.114.3 Entidad respondedora (SSF) .....	708
17.115 Procedimiento SelectRoute .....	708
17.115.1 Descripción general .....	708
17.115.2 Entidad invocadora (SCF).....	710
17.115.3 Entidad respondedora (SSF) .....	710

	Página
17.116 Procedimiento SendChargingInformation.....	712
17.116.1 Descripción general .....	712
17.116.2 Entidad invocadora (SCF).....	712
17.116.3 Entidad respondedora (SSF) .....	713
17.117 Procedimiento SendComponent.....	714
17.117.1 Descripción general .....	714
17.117.2 Entidad invocadora (SCF).....	714
17.117.3 Entidad respondedora (CUSF).....	715
17.118 Procedimiento SendFacilityInformation.....	715
17.118.1 Descripción general .....	715
17.118.2 Entidad invocadora (SCF).....	716
17.118.3 Entidad respondedora (SSF) .....	716
17.119 Procedimiento SendSTUI.....	717
17.119.1 Descripción general .....	717
17.119.2 Entidad invocadora (SCF).....	717
17.120 Procedimiento ServiceFilteringResponse.....	718
17.120.1 Descripción general .....	718
17.120.2 Entidad invocadora (SSF).....	718
17.120.3 Entidad respondedora (SCF).....	719
17.121 Procedimiento SpecializedResourceReport .....	720
17.121.1 Descripción general .....	720
17.121.2 Entidad invocadora (SRF).....	720
17.121.3 Entidad respondedora (SCF).....	720
17.122 Procedimiento SplitLeg.....	721
17.122.1 Descripción general .....	721
17.122.2 Entidad invocadora (SCF).....	721
17.122.3 Entidad respondedora (SSF) .....	721
17.123 Procedimiento StatusReport.....	722
17.123.1 Descripción general .....	722
17.123.2 Entidad invocadora (SSF).....	722
17.123.3 Entidad respondedora (SCF).....	723
17.124 Procedimiento TAnswer.....	723
17.124.1 Descripción general .....	723
17.124.2 Entidad invocadora (SSF).....	724
17.124.3 Entidad respondedora (SCF).....	725
17.125 Procedimiento TBusy .....	726
17.125.1 Descripción general .....	726

	Página
17.125.2 Entidad invocadora (SSF).....	726
17.125.3 Entidad respondedora (SCF).....	727
17.126 Procedimiento TDisconnect .....	728
17.126.1 Descripción general .....	728
17.126.2 Entidad invocadora (SSF).....	729
17.126.3 Entidad respondedora (SCF).....	729
17.127 Procedimiento TerminationAttempt.....	730
17.127.1 Descripción general .....	730
17.127.2 Entidad invocadora (SSF).....	731
17.127.3 Entidad respondedora (SCF).....	731
17.128 Procedimiento TermAttemptAuthorized.....	731
17.128.1 Descripción general .....	731
17.128.2 Entidad invocadora (SSF).....	732
17.128.3 Entidad respondedora (SCF).....	733
17.129 Procedimiento TMidCall.....	733
17.129.1 Descripción general .....	733
17.129.2 Entidad invocadora (SSF).....	734
17.129.3 Entidad respondedora (SCF).....	735
17.130 Procedimiento TNoAnswer.....	735
17.130.1 Descripción general .....	735
17.130.2 Entidad invocadora (SSF).....	736
17.130.3 Entidad respondedora (SCF).....	737
17.131 Procedimiento TSuspended.....	738
17.131.1 Descripción general .....	738
17.131.2 Entidad invocadora .....	738
17.131.3 Entidad respondedora (SCF).....	739
17.132 Procedimiento UpdateShadow .....	739
17.132.1 Descripción general .....	739
17.132.2 Entidad del suministrador (SDF) .....	740
17.132.3 Entidad del consumidor (SDF) .....	741

## **PARTE 4**

18 Servicios asumidos de las capas inferiores.....	743
18.1 Servicios asumidos de la TCAP .....	743
18.1.1 Procedimientos comunes.....	743
18.1.2 Interfaz SSF-SCF .....	754
18.1.3 Interfaz SCF-SRF.....	762

	Página
18.1.4 Interfaz SCF-CUSF .....	763
18.1.5 Interfaz SCF-SCF .....	766
18.1.6 Interfaz SCF-SDF .....	768
18.1.7 Interfaz SDF-SDF .....	771
18.2 Servicios asumidos de la SCCP .....	774
18.2.1 Procedimientos normales .....	774
18.2.2 Funciones de servicio de la SCCP .....	775
19 Seguridad de interfaz genérica de la red inteligente .....	778
19.1 Requisitos de seguridad de la interfaz .....	778
19.1.1 Confidencialidad de los datos .....	778
19.1.2 Integridad de los datos y autenticación del origen de los datos .....	779
19.1.3 Gestión de las claves .....	779
19.2 Procedimientos y algoritmos .....	779
19.2.1 Procedimientos de autenticación .....	779
19.2.2 Algoritmos y negociación SPKM .....	780
19.2.3 Autenticación mutua tridireccional .....	781
19.2.4 Asignación de credenciales .....	782
19.3 Correspondencia entre definiciones de flujos de información de seguridad y testigo .....	782
19.4 Definiciones de la FSM de seguridad .....	782
19.4.1 FSM de autenticación mutua bidireccional .....	782
19.4.2 FSM de autenticación mutua tridireccional .....	785
<b>PARTE 5</b>	
Anexo A.1 – Introducción a los modelos SDL del CS-1 y del CS-2 del protocolo de aplicación de la red inteligente (INAP) .....	789
A.1.1 Introducción .....	789
A.1.2 Ejemplo de interfuncionamiento de los procesos SDL de la SSF y la CCF .....	794
A.1.3 Ejemplo de establecimiento del servicio tripartito visto desde el entorno ....	796
Anexo A.2 – Diagramas de transición .....	799
A.2.1 Diagrama de transición de la asociación de segmentos de llamada .....	799
A.2.2 Diagrama de transición de segmentos de llamada .....	800
Anexo A.3 – Especificación en SDL de la SSF/CCF del CS-1 .....	814
<b>PARTE 6</b>	
Anexo A.4 – Especificación en SDL de las extensiones a la SSF/CCF del CS-2 .....	815

**PARTE 7**

Anexo A.5 – Especificación en SDL de la SRF del CS-2.....	816
Anexo A.6 – Especificación en SDL de la SSF de asistencia/desasistencia del CS-2.....	816
Anexo A.7 – Especificación en SDL de la CUSF del CS-2.....	816
Anexo A.8 – Especificación en SDL de la SCF del CS-2.....	816
Apéndice I – Fuente ASN.1 expandida .....	817
Apéndice II – Modelado de datos.....	846
II.1    Introducción.....	846
II.1.1    Objetivo y ámbito.....	846
II.1.2    Hipótesis.....	846
II.2    Esquemas del árbol de información de directorio (DIT, <i>directory information tree</i> )	847
II.2.1    Árbol de información del directorio de la Recomendación X.500 .....	847
II.2.2    Clases de objetos .....	850
II.2.3    Tipos de atributos.....	851
II.2.4    Definición de la estructura del árbol de información de directorio (DIT) ....	851
Apéndice III – Ejemplos de algoritmos SPKM para el CS-2 de la red inteligente.....	854
III.1    Consideraciones generales.....	854
III.2    Algoritmo de integridad (I-ALG).....	854
III.2.1    Ejemplo–1 .....	854
III.2.2    Ejemplo–2 .....	854
III.2.3    Ejemplo–3 .....	855
III.2.4    Ejemplo–4 .....	855
III.3    Algoritmo de confidencialidad (C-ALG).....	855
III.3.1    Ejemplo–1 .....	856
III.4    Algoritmo de establecimiento de clave (K-ALG) .....	856
III.4.1    Ejemplo–1 .....	856
III.4.2    Ejemplo–2 .....	856
III.4.3    Ejemplo–3 .....	856
III.5    Función unidireccional (O-ALG) para el algoritmo de derivación de subclaves.....	856
III.5.1    Ejemplo-1 .....	857



## **Recomendación Q.1228**

### **RECOMENDACIÓN SOBRE INTERFACES PARA EL CONJUNTO DE CAPACIDADES 2 DE RED INTELIGENTE**

*(Ginebra, 1997)*

## **PARTE 5**

### **ANEXO A.1**

#### **Introducción a los modelos SDL del CS-1 y del CS-2 del protocolo de aplicación de la red inteligente (INAP)**

##### **A.1.1 Introducción**

Los modelos SDL incluidos en este anexo constituyen una referencia normativa y son considerados parte de la presente Recomendación.

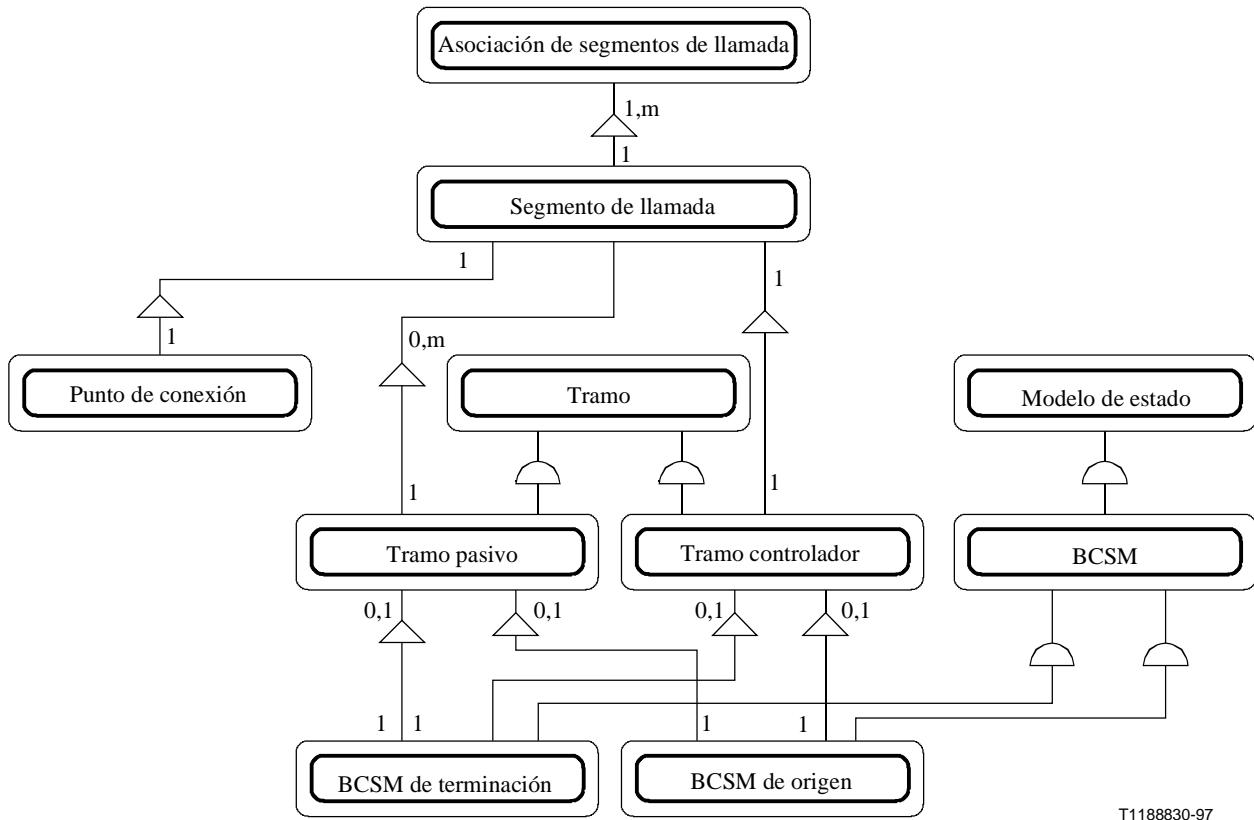
Los diagramas SDL del CS-1 de red inteligente no se ajustan con precisión al comportamiento del INAP especificado en esta Recomendación, ya que algunos cambios se han efectuado por conveniencia de la modelación. El comportamiento del INAP especificado en esta Recomendación está correctamente modelado por la combinación de las especificaciones del CS-1 y del CS-2 de red inteligente que figuran en este anexo.

Los modelos CS-1 y CS-2 de red inteligente del INAP se especifican con el SDL 92 de un modo orientado al objeto. La especificación del CS-2 de red inteligente hereda el CS-1 de red inteligente y especifica únicamente la diferencia entre el CS-1 y el CS-2. Por consiguiente, las arquitecturas del CS-1 y del CS-2 de red inteligente son aproximadamente las mismas.

El modelo SDL especifica con precisión y sin ambigüedades el comportamiento y el interfuncionamiento entre las distintas entidades funcionales: CS, CSA, FSM de la SSF, BCSM. Las estructuras de datos no se especifican por completo. Están incluidas en esta Recomendación y se efectúan referencias adecuadas en el modelo SDL.

El modelo proporciona una plataforma para emulación de servicios y el desarrollo de casos de prueba basados en servicios de la red inteligente.

La figura A.1-1 muestra el modelo de información CS-1/CS-2 de red inteligente. En una asociación de segmentos de llamada existe uno, o muchos o ningún segmento de llamada. Hay una correspondencia biunívoca entre el segmento de llamada y el punto de conexión, y así sucesivamente. Hay dos objetos de tipo BCSM, el BCSM de origen y el BCSM de terminación.



**Figura A.1-1/Q.1228 – Modelo de información CS-1/CS-2 de red inteligente**

La figura A.1-2 muestra el modelo SDL de las SSF/CCF del CS-1 de red inteligente. El CS-2 de red inteligente equivalente, que aparece en la figura A.1-5, hereda la mayoría de los ítems del modelo CS-1 de red inteligente y los mejora. Los objetos del modelo de información de la figura A.1-1 se corresponden con el modelo SDL de la figura A.1-2 del modo como se indica a continuación.

El objeto asociación de segmentos de llamada es un tipo de proceso del modelo SDL. La asociación de segmentos de llamada gestiona:

- la creación de segmentos de llamada; y
- el diálogo con la SCF.

El objeto segmento de llamada se divide en dos tipos de procesos en el modelo SDL: segmento de llamada y FSM de la SSF. El tipo de proceso segmento de llamada gestiona:

- los identificadores de los tramos (visión de la conexión). Esta estructura de datos modela el objeto punto de conexión;
- la creación de los O-BCSM y los T-BCSM;
- la creación de la FSM de la SSF;
- el filtrado de puntos de detección;
- el procesamiento de operaciones de red inteligente orientadas a la visión de la conexión.

El proceso de segmento de llamada define estados de visión de la conexión para el CS y no debe confundirse con la SSF para el CS indicado en 11.5.2 ni con el planteamiento estados de visión de la conexión (CVS, *connection view state*) de la Recomendación Q.1224. El motivo es que en la Recomendación Q.1224 el planteamiento CVS proporciona nombres explícitos para las combinaciones de estados de los CS asociados. En la Recomendación Q.1228 esto llevaría a un número infinito de estados de asociación de segmentos de llamada (CSA, *call segment association*)

dado que la Recomendación Q.1224 representa únicamente un subconjunto muy pequeño de los estados de protocolo. En cambio, en los SDL, los estados de cada CS reciben un nombre (por ejemplo, parte 2 estable) y la CSA está representada como un conjunto de CS asociados, posibles en distintos estados.

El tipo de proceso FSM de la SSF gestiona:

- el procesamiento de las operaciones de red inteligente;
- el tratamiento de puntos de detección (EDP y TDP).

Para centralizar el proceso del punto de detección (DP, *detection point*) y simplificar así el modelo, se crea la FSM de la SSF incluso si no puede armarse ningún punto de detección de activador (TDP, *trigger detection point*) para la llamada.

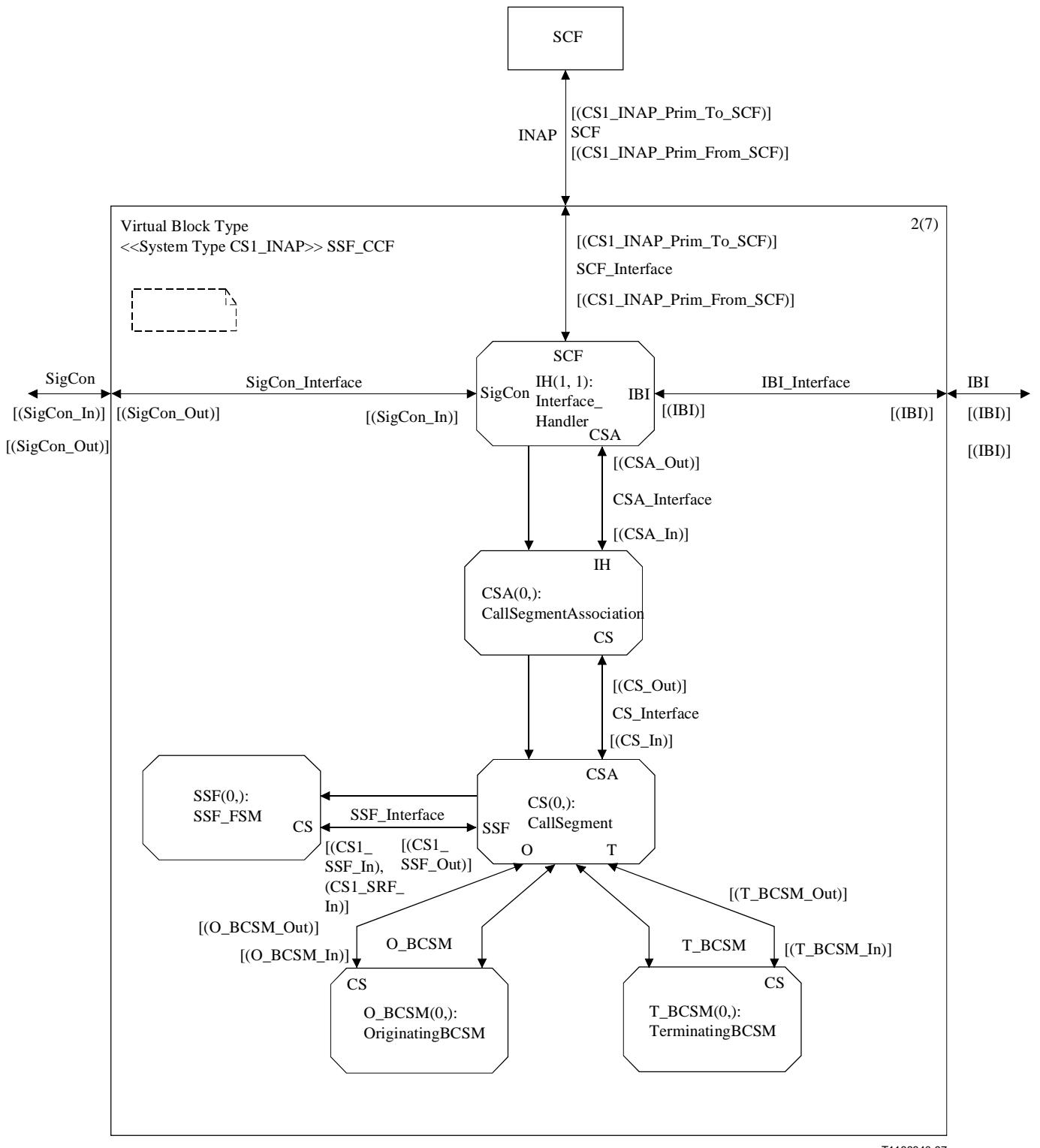
La FSM de la SSF representada por los SDL es diferente de la FSM para el CS descrita en 11.5.2. La FSM de la SSF en el SDL tiene tres estados y trata la interacción de usuario en el modo supervisión.

El objeto punto de conexión es modelado por una estructura de datos como la ya descrita.

El objeto tramo es también una estructura de datos del tipo de proceso segmento de llamada. Es idéntico a las estructuras de datos del tramo pasivo y el tramo controlador. La estructura de datos contiene el estado del tramo y un puntero al BCSM conectado al tramo.

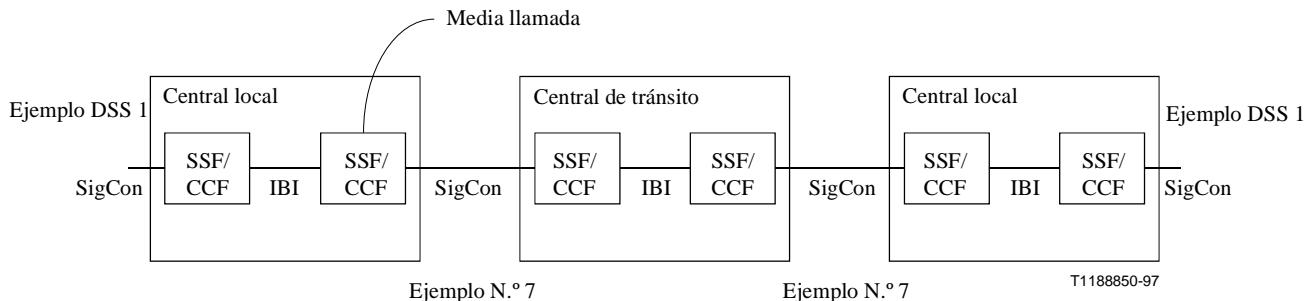
El BCSM de terminación y el BCSM de origen son objetos de la clase BCSM en el modelo de información. El modelo SDL no se construye exactamente de esta manera. Dado que el O-BCSM y el T-BCSM son significativamente distintos, son modelados en dos tipos de proceso completamente independientes.

El modelo SDL contiene un tipo de proceso adicional: el manipulador de interfaz (IH, *interface handler*). El manipulador de interfaz es el gestor permanente de la función control de llamada. Cuando comienza la interpretación de la especificación SDL, el IH es el único proceso que existe. A continuación, en el curso del establecimiento de una llamada, el IH crea la asociación segmento de llamada, tras haber recibido del entorno los mensajes adecuados. Además, manipula el diálogo con la SCF y transmite las primitivas entre la interfaz del control de señalización y las CSA y entre la interfaz de la SCF y las CSA.



**Figura A.1-2/Q.1228 – Modelo SDL del CS-1 de red inteligente del INAP, visión de media llamada**

T1188840-97

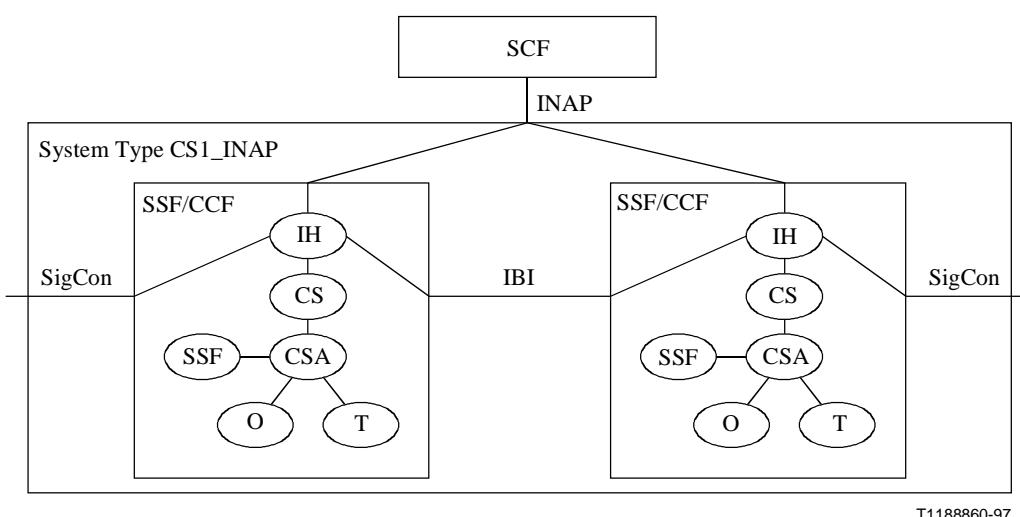


**Figura A.1-3/Q.1228 – Ejemplo general de escenario para INAP que muestra únicamente la SSF/CCF**

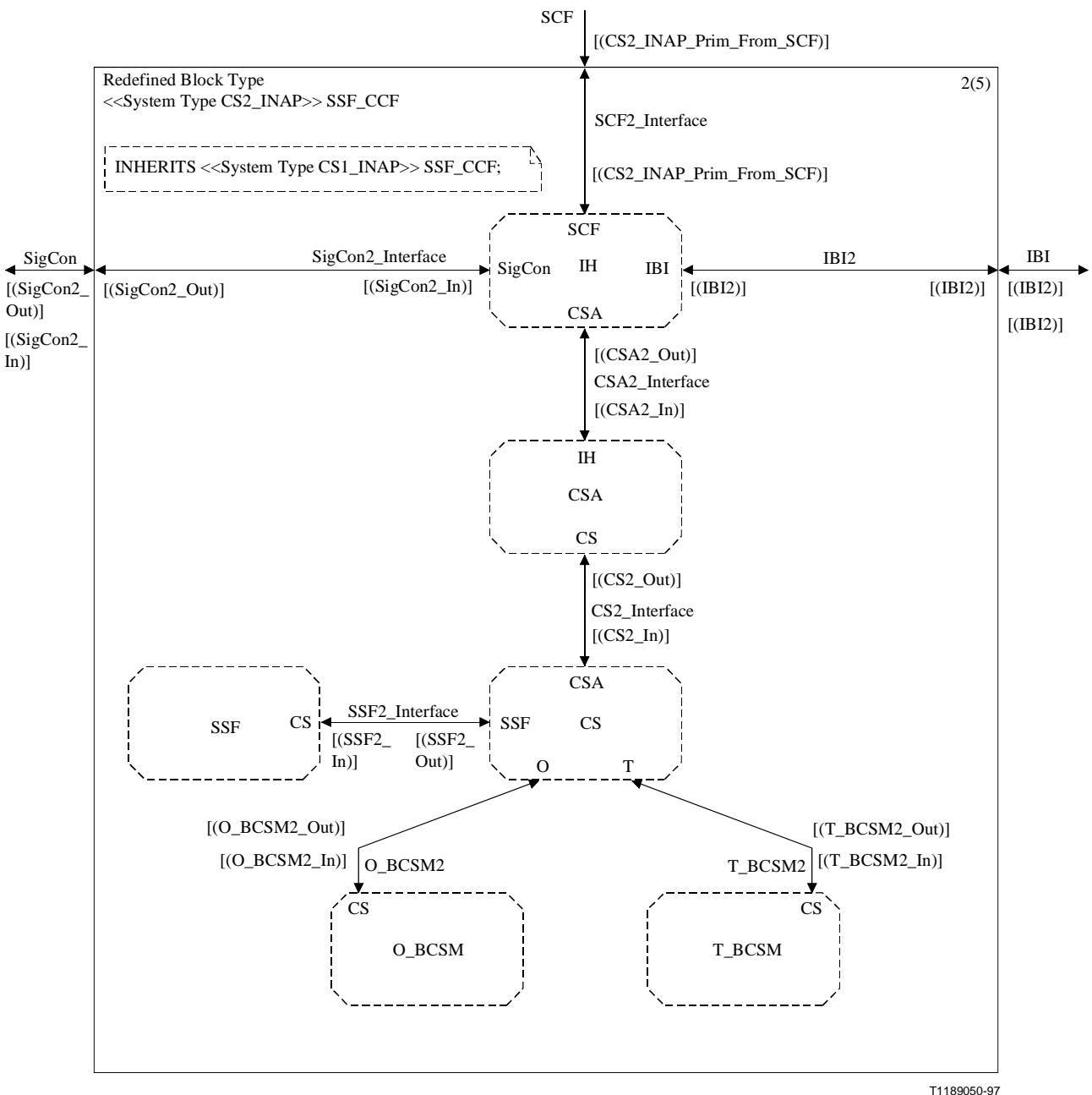
La figura A.1-3 muestra un ejemplo general de escenario para red inteligente relacionado con el modelo SDL. La interfaz IBI es una interfaz BCSM interna entre dos medias llamadas.

El modelo SDL completo consta de dos medias llamadas, como indica la figura A.1-4. La figura A.1-2 es una mejora del bloque SSF/CCF. En ella aparece una media llamada bajo el control de la SCF, incluidas las funciones de conmutación de servicio y de control de llamada. Para que el modelo funcione, es necesario duplicar el bloque SSF\_CCF para obtener dos medias llamadas. De este modo, puede efectuarse la simulación de la funcionalidad completa del interfuncionamiento entre el O-BCSM y el T-BCSM.

El modelo tiene tres interfaces. La interfaz SigCon podría ser, por ejemplo, una interfaz DSS 1. Los mensajes por SigCon son abstractos y deben hacerse corresponder con un protocolo real, por ejemplo, DSS 1, en un caso concreto. La interfaz IBI es completamente interna a un conmutador. La interfaz INAP con la SCF es la que tiene mensajes INAP normalizados.



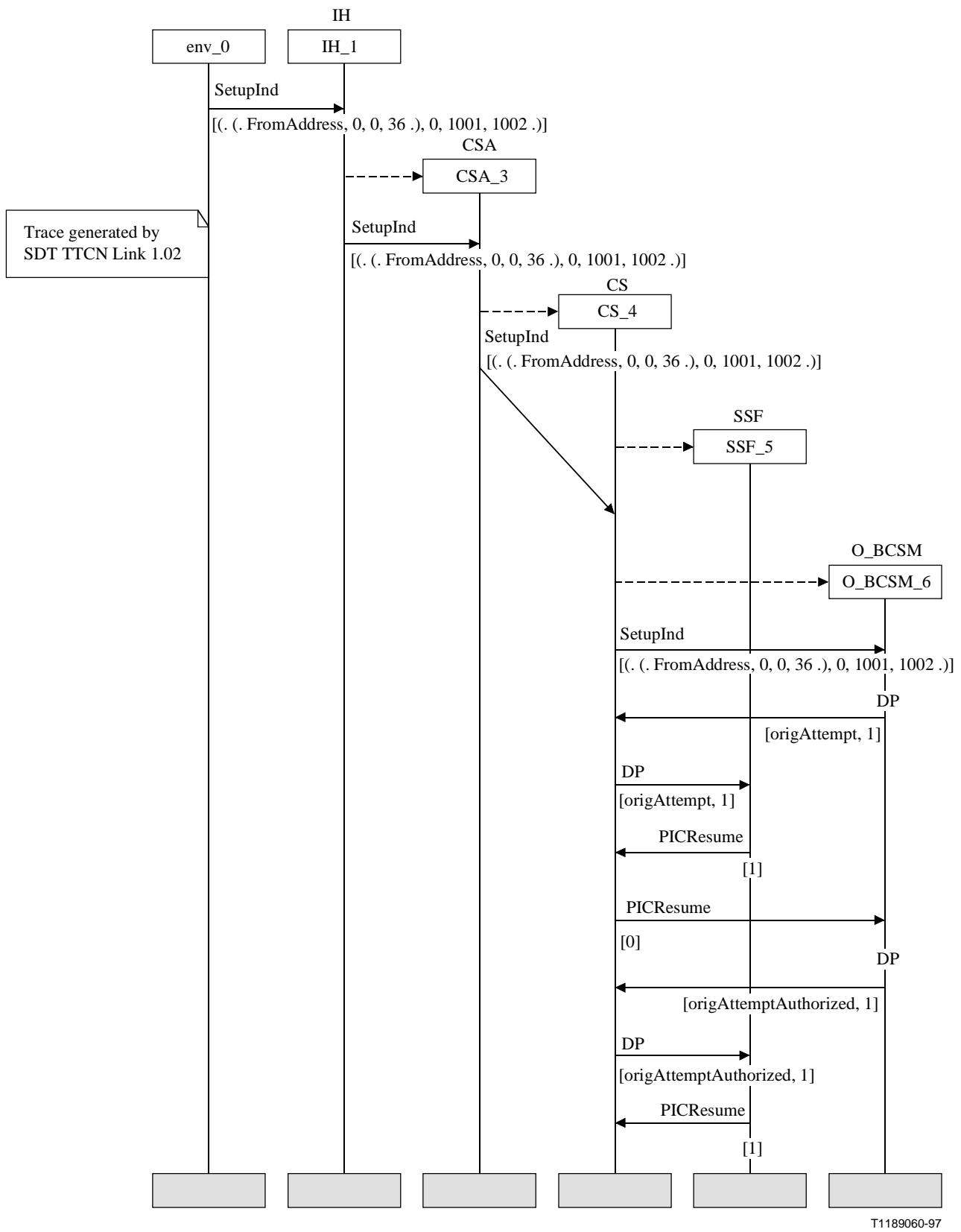
**Figura A.1-4/Q.1228 – Dos medias llamadas**



**Figura A.1-5/Q.1228 – Modelo del CS-2 de red inteligente del INAP que hereda el CS-1 de red inteligente**

### A.1.2 Ejemplo de interfuncionamiento de los procesos SDL de la SSF y la CCF

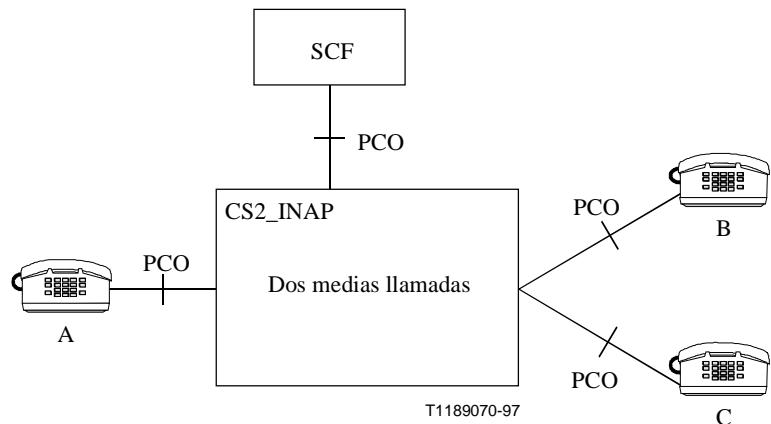
El siguiente gráfico de secuencia de mensajes (MSC, *message sequence chart*) muestra el interfuncionamiento del entorno, IH, CSA, CS, SSF y O\_BCSM después de que se recibe una SetupInd (indicación de establecimiento) a través de la interfaz SigCon. Por razones de brevedad, se muestran únicamente los primeros intercambios de señales. Véase la figura A.1-6.



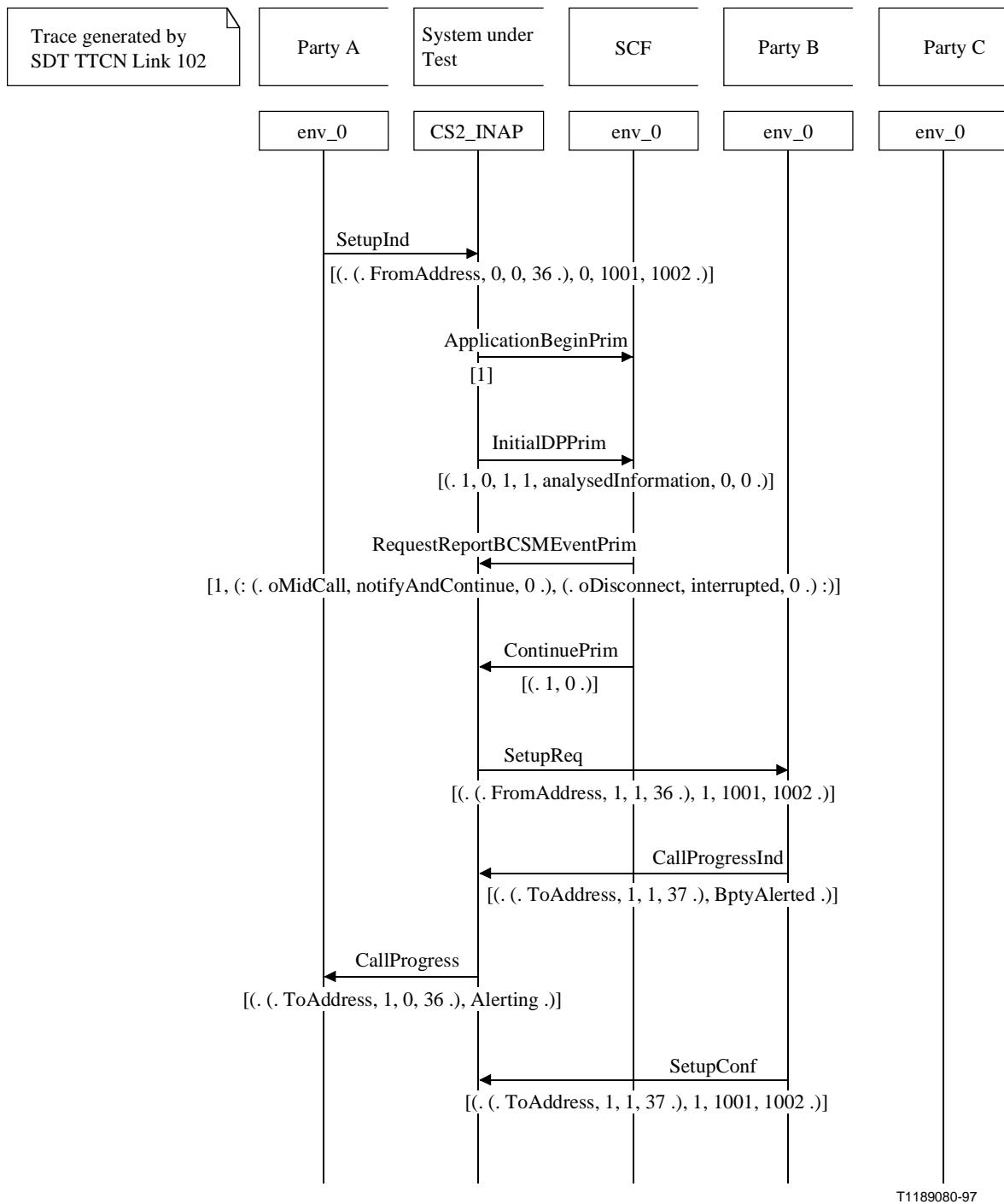
**Figura A.1-6/Q.1228 – Gráfico de secuencia de mensajes (MSC) para el interfuncionamiento de los procesos SDL de la SSF y la CCF**

### A.1.3 Ejemplo de establecimiento del servicio tripartito visto desde el entorno

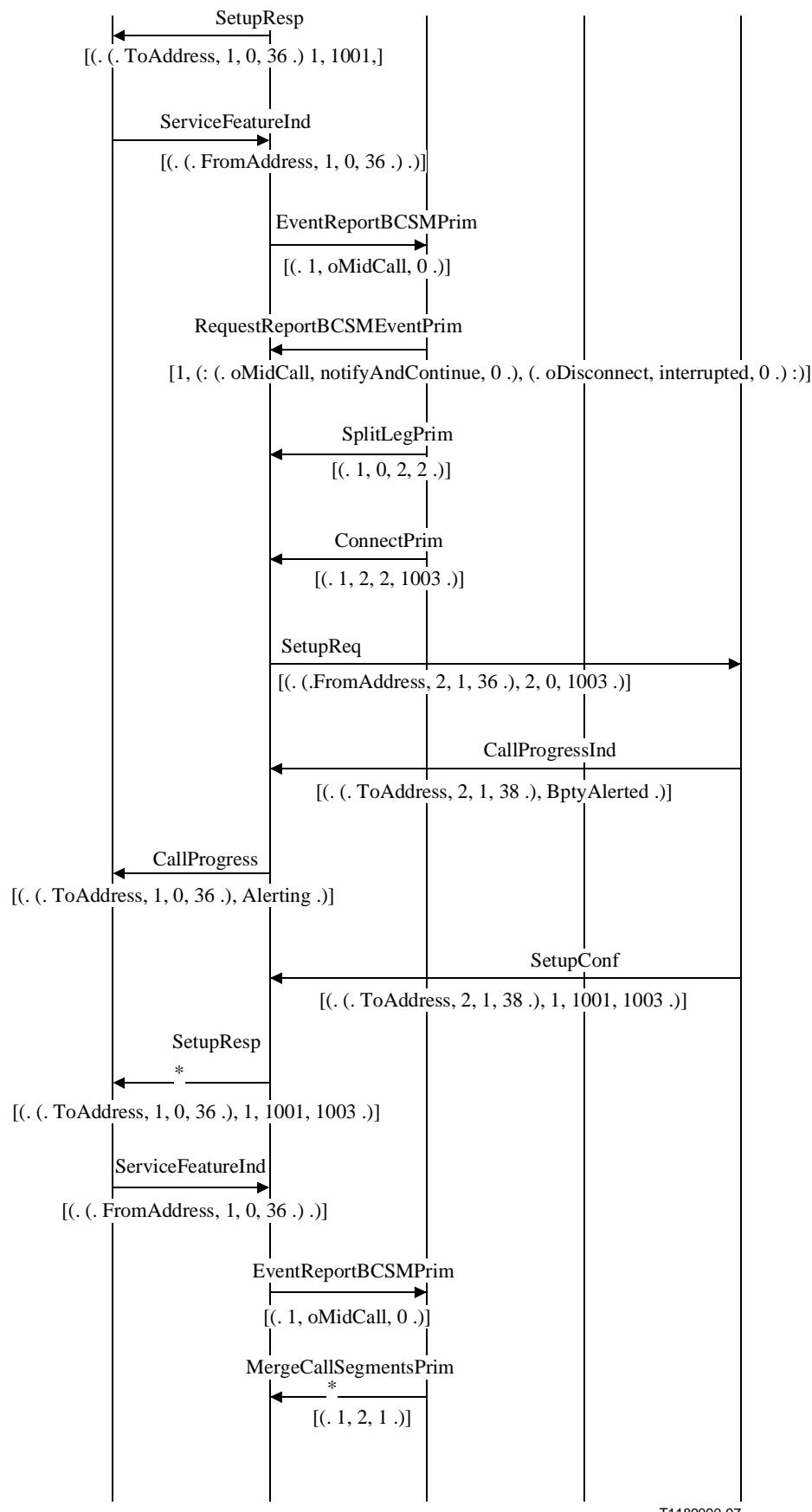
El siguiente gráfico de secuencia de mensajes (MSC) muestra un ejemplo de interfuncionamiento entre el modelo SDL y el entorno durante un servicio tripartito. El MSC se ha diseñado de conformidad con la arquitectura de la figura A.1-4. En el MSC no se muestra el intercambio de mensajes internos del modelo CS-2 del INAP de dos medias llamadas. Participan tres centrales locales: A, B y C. Desde el punto de vista del MSC, ellas son entornos, junto con la SCF. Para mayor claridad, cada uno de estos entornos se coloca en un eje de instancia de proceso distinto, correspondiendo cada uno de ellos a un punto de control y observación (PCO, *point of control and observation*). Véanse las figuras A.1-7 y A.1-8.



**Figura A.1-7/Q.1228 – Establecimiento del servicio tripartito  
– Dos medias llamadas**



**Figura A.1-8/Q.1228 (hoja 1 de 2) – Gráfico de secuencia de mensajes (MSC) – Servicio tripartito**



T1189090-97

**Figura A.1-8/Q.1228 (hoja 2 de 2) – Gráfico de secuencia de mensajes (MSC) – Servicio tripartito**

## ANEXO A.2

### Diagramas de transición

#### A.2.1 Diagrama de transición de la asociación de segmentos de llamada

##### A.2.1.1 Definición de estados y transiciones

Los estados identificados para la visión de conexión de la asociación de segmentos de llamada (CSACV) son los siguientes:

- Reposo/nulo:** La condición "reposo" indica que no existe ninguna instancia de CSACV y que puede ser creada; esta condición se produce cuando se suprime una instancia de CSACV particular. El estado "nulo" representa la condición en la cual se crea una asociación de segmentos de llamada sin ningún segmento de llamada y un diálogo se abre con la lógica de servicio.
- Un segmento:** Este estado representa una asociación de segmentos de llamada que contiene un segmento de llamada.
- Multisegmentos:** Este estado representa una asociación de segmentos de llamada que contiene múltiples segmentos de llamada.

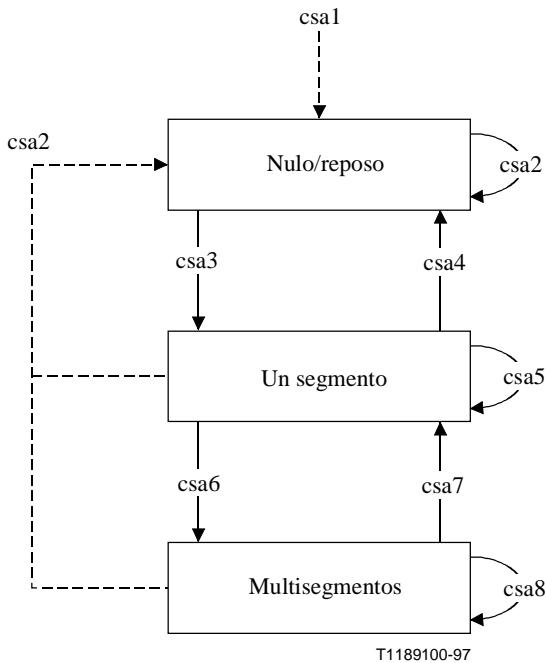
##### A.2.1.2 Procedimientos funcionales generales

###### A.2.1.2.1 Puesta en cola de eventos del BCSM al recibir una operación RRBE en la CSA

Los eventos armados deben almacenarse en una cola asociada a la instancia de CSA para los eventos recibidos en una operación RRBCSM para la que no se crea ningún tramo pasivo. Cuando posteriormente se envía una operación Connect (conexión) o InitiateCallAttempt (iniciación de intento de llamada), los eventos armados se recuperarán y transferirán a la instancia de CS existente o recientemente creada en el momento en que se crea el tramo. Los eventos armados tienen que almacenarse en el nivel de la CSA y no en el CS creado inicial porque puede ocurrir que se suprima ese CS así como la cola asociada.

###### A.2.1.3 Modelo de estados finitos para la visión de conexión de la asociación de segmentos de llamada (CSACV)

En la figura A.2-1 se indica el modelo de estados finitos para la visión de conexión de la asociación de segmentos de llamada (CSACV, *call segment association connection view*).



**Figura A.2-1/Q.1228 – Modelo de estados finitos para la visión de conexión de la asociación de segmentos de llamada**

El diagrama de estados de la SSF de la CSACV contiene los siguientes eventos:

- csa1: Crear asociación de segmentos de llamada.
- csa2: Suprimir asociación de segmentos de llamada.
- csa3: SetupInd (indicación de establecimiento), SetupReqInd (indicación de petición de establecimiento), InitiateCallAttempt (iniciación de intento de llamada).
- csa4: ReleaseCall (liberación de llamada).
- csa5: Reconnect (reconexión), DisconnectLeg (desconexión de tramo).
- csa6: SplitLeg (división de tramo), InitiateCallAttempt (iniciación de intento de llamada).
- csa7: ReleaseCall (liberación de llamada), MergeCallSegments (fusión de segmentos de llamada).
- csa8: SplitLeg (división de tramo), InitiateCallAttempt (iniciación de intento de llamada), MergeCallSegments (fusión de segmentos de llamada), Reconnect (reconexión), DisconnectLeg (desconexión de tramos), MoveLeg (desplazamiento de tramo).

## A.2.2 Diagrama de transición de segmentos de llamada

### A.2.2.1 Definiciones de estados y transiciones

Los estados identificados para la visión de conexión de segmentos de llamada (CSCV) son los siguientes:

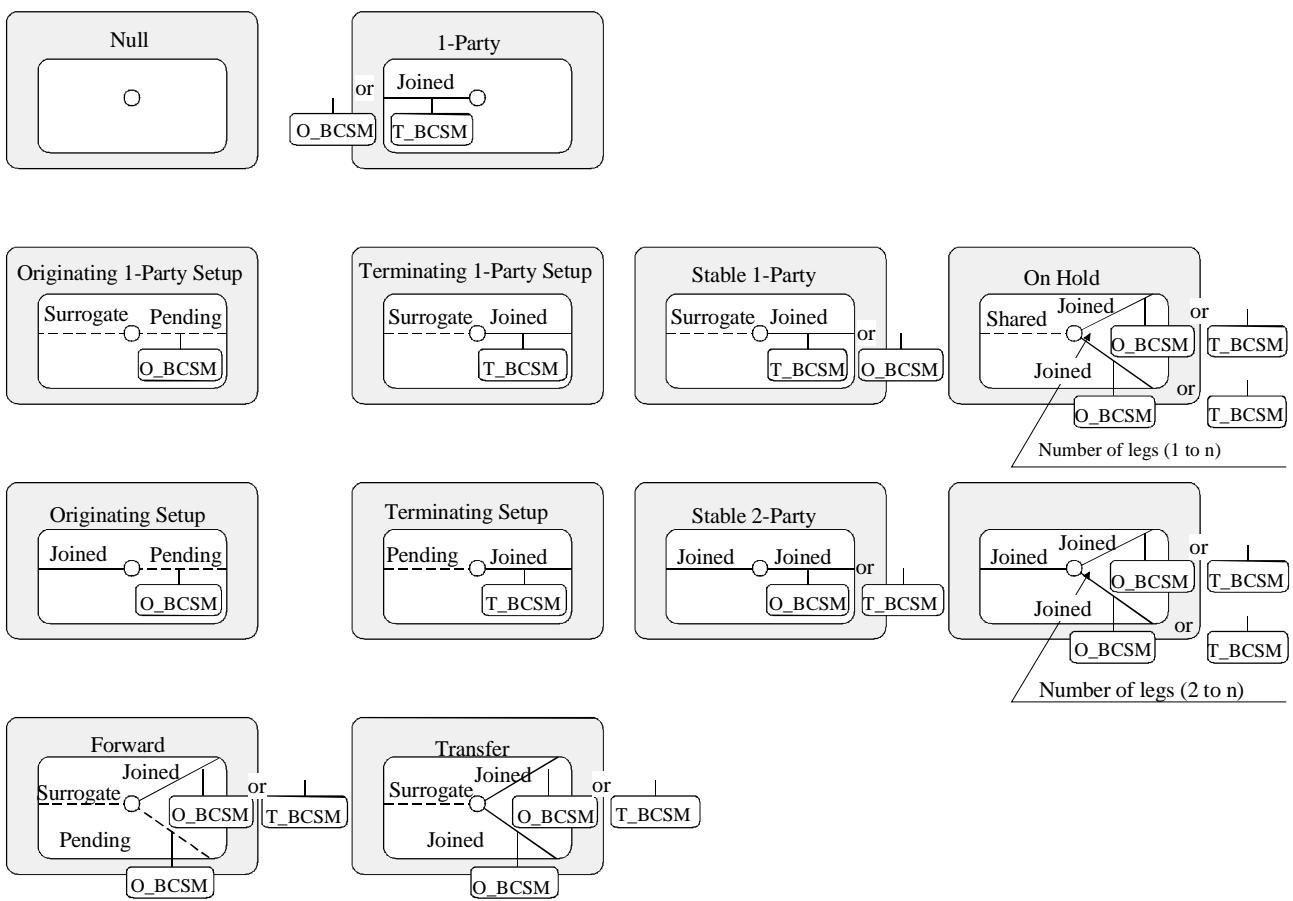
- Reposo/Nulo (Idle/Null):** La condición "receso" (idle) indica que no existe ninguna instancia de CSCV y que puede ser creada; esta condición se produce cuando se suprime una instancia de CSCV particular. El estado "nulo" (null) representa la condición en la cual el procesamiento de llamada no está activo y no existe ningún tramo controlador ni tramo pasivo conectado al punto de conexión.

<b>Establecimiento de origen (Originating Setup):</b>	Este estado representa un servicio bipartito de origen en la fase establecimiento con el tramo controlador en estado "unido" (joined) y un tramo pasivo "pendiente" (pending) con un O_BCSM asociado.
<b>Establecimiento de terminación (Terminating Setup):</b>	Este estado representa un servicio bipartito de terminación en la fase establecimiento con el tramo controlador en estado "pendiente" (pending) y un tramo pasivo "unido" (joined) con un T_BCSM asociado.
<b>Bipartito estable (Stable 2-Party):</b>	Este estado representa un servicio bipartito estable o de liberación, y es una llamada de origen o bien una llamada de terminación desde la perspectiva del usuario controlador con un tramo controlador "unido" (joined) y un tramo pasivo "unido" (joined) con un O_BCSM o bien un T_BCSM asociado a él.
<b>Unipartito (1-Party):</b>	Este estado representa un servicio unipartito que se origina con el tramo controlador en estado "unido" (joined) y ningún tramo pasivo. El BCSM está asociado al tramo controlador puesto que ningún tramo pasivo está conectado al punto de conexión (CP).
<b>Establecimiento unipartito de origen (Originating 1-Party Setup):</b>	Este estado representa un servicio unipartito que se origina en nombre de la red con el tramo controlador en estado "subrogado" (surrogate) y un estado de tramo pasivo "pendiente" (pending) al que está asociado un O_BCSM.
<b>Establecimiento unipartito de terminación (Terminating 1-Party Setup):</b>	Este estado representa un servicio unipartito que se termina en nombre de la red con el tramo controlador en el estado "subrogado" (surrogate) y un estado de tramo pasivo "unido" (joined) al que está asociado un T_BCSM.
<b>Unipartito estable (Stable 1-Party):</b>	Este estado representa un servicio unipartito originado en nombre de la red con el tramo controlador en estado "subrogado" (surrogate) y un tramo pasivo "unido" (joined) con un O_BCSM o bien un T_BCSM asociado a él, que está en una fase estable o de liberación.
<b>Reenvío (Forward):</b>	Este estado representa una llamada reenviada. El procesamiento de la llamada de los tramos originalmente pasivos de la parte llamante está en una fase de establecimiento de llamada de origen o de terminación/fase estable, mientras que el procesamiento de llamadas del tramo pasivo al que se ha reenviado la llamada está en una fase de establecimiento de llamada de origen. El segmento de llamada tiene un tramo controlador "subrogado" (surrogate) con uno o dos tramos pasivos "unidos" (joined) de la parte llamante con un O_BCSM o bien un T_BCSM asociado y un tramo pasivo al que se ha reenviado la llamada "pendiente" (pending) con un O_BCSM asociado.
<b>Transferencia (Transfer):</b>	Este estado representa una llamada transferida. La llamada entre los tramos pasivos implicados está en la fase estable o de liberación.

**Retención (On Hold):** Este estado representa una llamada retenida, en la cual el usuario controlador ha puesto en retención las partes distantes implicadas y participa en otra llamada.

**Multipartito estable (Stable Multi-Party):** Este estado representa un servicio multipartito estable o de liberación.

Los estados identificados para la visión de conexión de segmentos de llamada (CSCV, *call segment connection view*) se indican en la figura A.2-2 siguiente:



**Figura A.2-2/Q.1228 – Estados de la CSCV**

## A.2.2.2 Procedimientos funcionales generales para la CSCV

Se aplicarán los siguientes principios generales:

### A.2.2.2.1 Indicación del tipo de BCSM

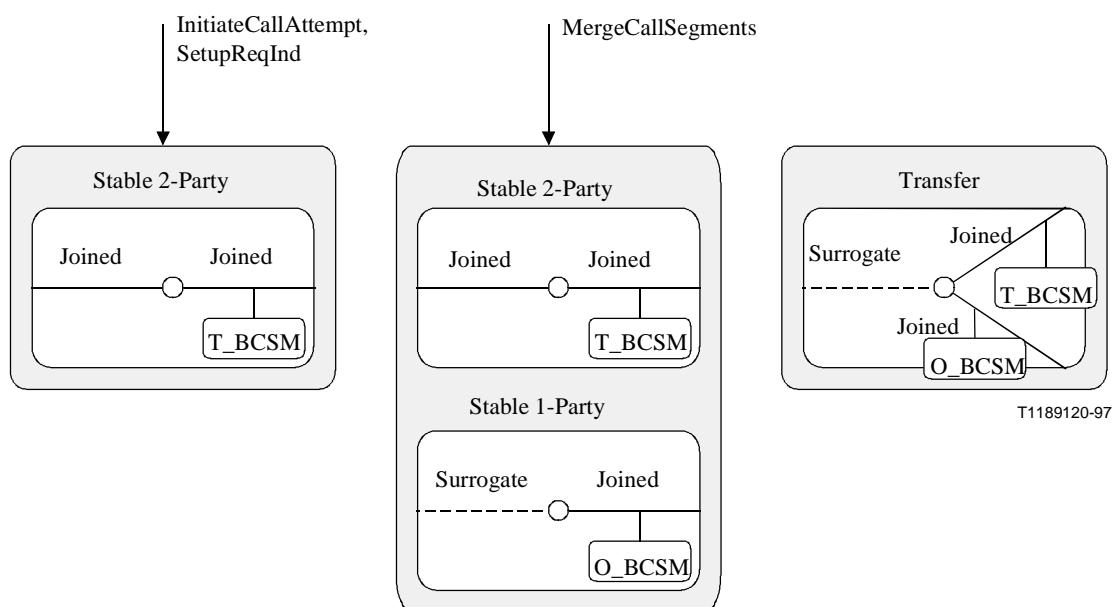
Se asignará un atributo "tipo BCSM" a un tramo controlador en el momento en que éste es creado, indicando si el tramo controlador pertenece a un BCSM de origen o de terminación, es decir, una indicación del tipo de BCSM en el que se produjo la activación de la llamada.

Cuando en un CS se deje un tramo controlador, el atributo "tipo BCSM" determinará qué BCSM se aplicará. Las condiciones del tramo, es decir los EDP armados, el ApplyChargingReport (informe de aplicación de tarificación) pendiente, la EventNotificationCharging (notificación de evento de

tarificación) pendiente y el CallInformationReport (informe de información de llamada) pendiente se aplicarán también en el mismo tramo después de la creación de una instancia de BCSM [por ejemplo, después de una SplitLeg, (división de tramo)]. Ésta última implica que si no se armara ningún EDP ni ningún informe para el tramo en relación con el tipo de BCSM creado, se efectuaría un armado explícito utilizando, por ejemplo, la operación RequestReportBCSMEvent (petición de informe de evento BCSM), si fuese necesario.

Para obtener una completa armonización con el diseño del programa de procesamiento de lógica de servicio (SLP) en la SCF, no se establecerá ninguna correspondencia de eventos entre eventos O\_BCSM y eventos T\_BCSM, y viceversa. Esto significa que el punto de detección (DP) en el cual debe suspenderse el procesamiento de la llamada seleccionará en la instancia de BCSM asociada el estado real de la llamada considerado desde el punto de vista de los tramos; por ejemplo, si existe un "PIC de llamada activa", la instancia de BCSM se pondrá en el MidCall DP (DP de mitad de llamada) del estado BCSM activo cuando el tramo se deja en un CS después de, por ejemplo, una operación DisconnectLeg (desconexión de tramo) o SplitLeg (división de tramo). Para las instancias de O\_BCSM y T\_BCSM se señala lo siguiente:

- Cuando la instancia de O\_BCSM se pone en O\_Mid\_Call del estado O\_Active:
  - Es posible una transición a Collect\_Information, Analyse\_Information o Select\_Route mediante el envío de las operaciones apropiadas desde la SCF. Esto permite admitir nuevas llamadas antes de la liberación de conformidad con las transiciones BCSM ampliadas descritas en la Recomendación Q.1224.
- Cuando la instancia de T\_BCSM se pone en T\_Mid\_Call del estado T\_Active:
  - Para permitir el establecimiento de una conexión de llamada de origen a otra parte llamada de terminación, puede enviarse la operación ContinueWithArgument seguida de las operaciones InitiateCallAttempt y MergeCallSegments desde la SCF a fin de efectuar una "llamada de transferencia" (véase la figura A.2-3). Cabe observar que la transferencia de una llamada en este caso en que el tramo de control tiene el atributo "tipo BCSM" de terminación sólo es posible en una central de nodo servidora (no en una central de tránsito).



**Figura A.2-3/Q.1228 – Llamada de transferencia en T\_BCSM**

En los casos citados, el CS se pondrá en el estado unipartito con el estado SSF\_FSM asociado en "Espera de instrucciones" en el caso en que el tramo unido controlador se deje en un CS debido a una operación SCF (DisconnectLeg, SplitLeg, MoveLeg, etc.). Para reanudar desde los puntos de detección O\_Mid\_Call y T\_Mid\_Call los estados PIC O\_Active y T\_Active del BCSM asociados, puede ser enviada por la lógica de servicio una operación ContinueWithArgument.

Cabe destacar que no existe el problema de "trenzar" un tramo controlador "subrogado" como ocurre con un tramo de control "unido", porque se supone que un tramo subrogado "trenzado" se destruirá debido al hecho de que ningún trayecto vocal está asociado a ese tramo controlador.

#### A.2.2.2.2 Reglas de tratamiento de eventos

Se aplicarán las siguientes reglas de precedencia inter-BCSM para informar a la SCF de los eventos que deben señalarse para el tramo controlador:

- Sólo se comunicará una vez un evento a la SCF con independencia del número de tramos pasivos por CS conectados al tramo controlador (a través del CP).
- Un atributo "tipo BCSM" en el tramo controlador indica si este pertenece a un BCSM de origen o a un BCSM de terminación. El atributo "tipo BCSM" se fija de modo que refleje el tipo de BCSM (de origen o de terminación) en el que se produjo la activación o evento en el momento en que se creó la CS.
- Si el tramo controlador pertenece a un BCSM de origen, se informa como un evento de origen.
- Si el tramo controlador pertenece a un BCSM de terminación, se informa como un evento de terminación.

#### A.2.2.2.3 Creación de O\_T\_BCSM en base al atributo "tipo BCSM"

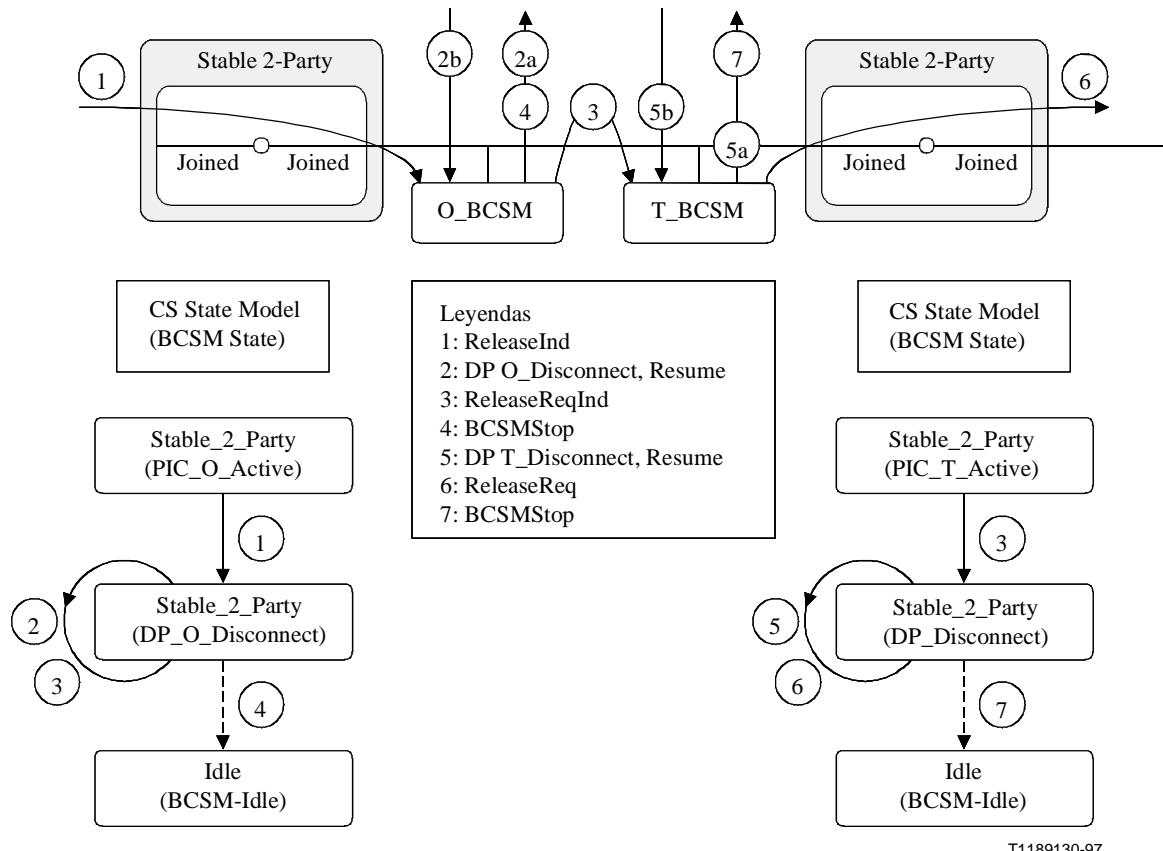
Se asigna una instancia de BCSM a un CS en el que sólo queda un tramo controlador (estado "unido" del tramo) para supervisar el tramo. Se utiliza el atributo "tipo BCSM" fijado en el tramo controlador para reflejar el tipo de BCSM (de origen o de terminación) que debe asignarse. La FSM de la SSF correspondiente para el CS se pondrá en el estado "Espera de instrucciones" cuando queda un tramo en un CS debido a una operación CPH (por ejemplo, DisconnectLeg) y se suspende el procesamiento de la llamada en el DP mitad de llamada del estado activo en el BCSM asociado.

Cuando el tramo controlador está conectado al tramo pasivo, no se conectarán ninguna instancia de BCSM al tramo controlador. La instancia de BCSM que estaba conectada al tramo controlador desaparece en el caso en que se desplace el tramo pasivo (importado) al segmento de llamada (por ejemplo, operación MergeCallSegments). Si se crea un nuevo tramo pasivo (por ejemplo, operación Connect) dentro del CS, la instancia de BCSM existente se conecta al tramo pasivo.

Una exportación de un tramo debida a la ejecución de una operación MoveLeg, SplitLeg o MergeCallSegments para un tramo pasivo sólo puede ejecutarse cuando el O\_BCSM está por lo menos en el PIC Send\_Call y para el T\_BCSM en el PIC T\_Active. Esto hace que no puedan ejecutarse estas operaciones cuando el tramo pasivo está en el estado pendiente.

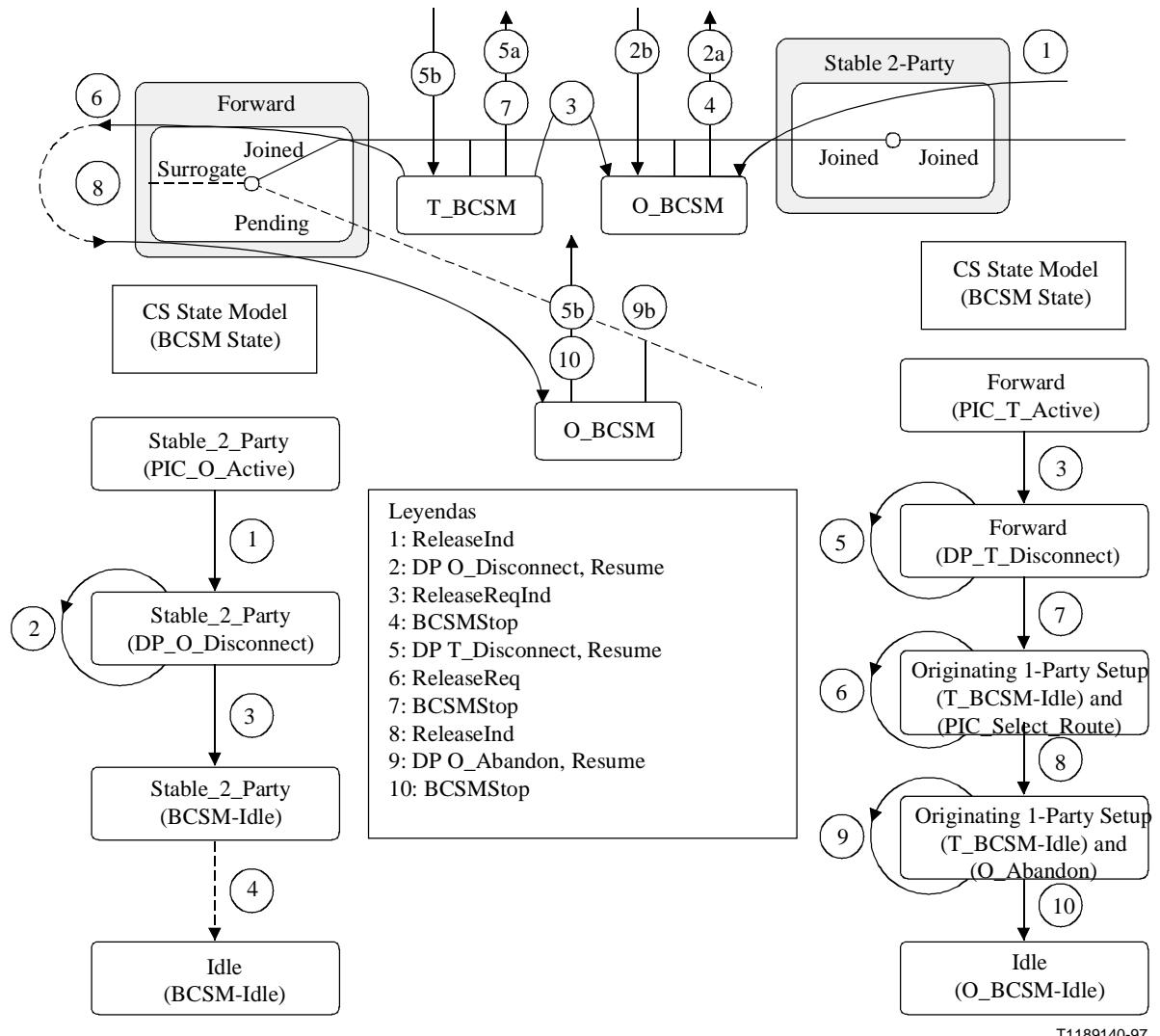
#### A.2.2.2.4 Liberación de una llamada/conexión

En la figura A.2-4 se indica la liberación de una llamada/conexión para una llamada normal cuando ambos segmentos de la llamada están en el estado "bipartito estable" (Stable\_2\_Party) (con el O\_BCSM y el T\_BCSM asociados en el estado activo).



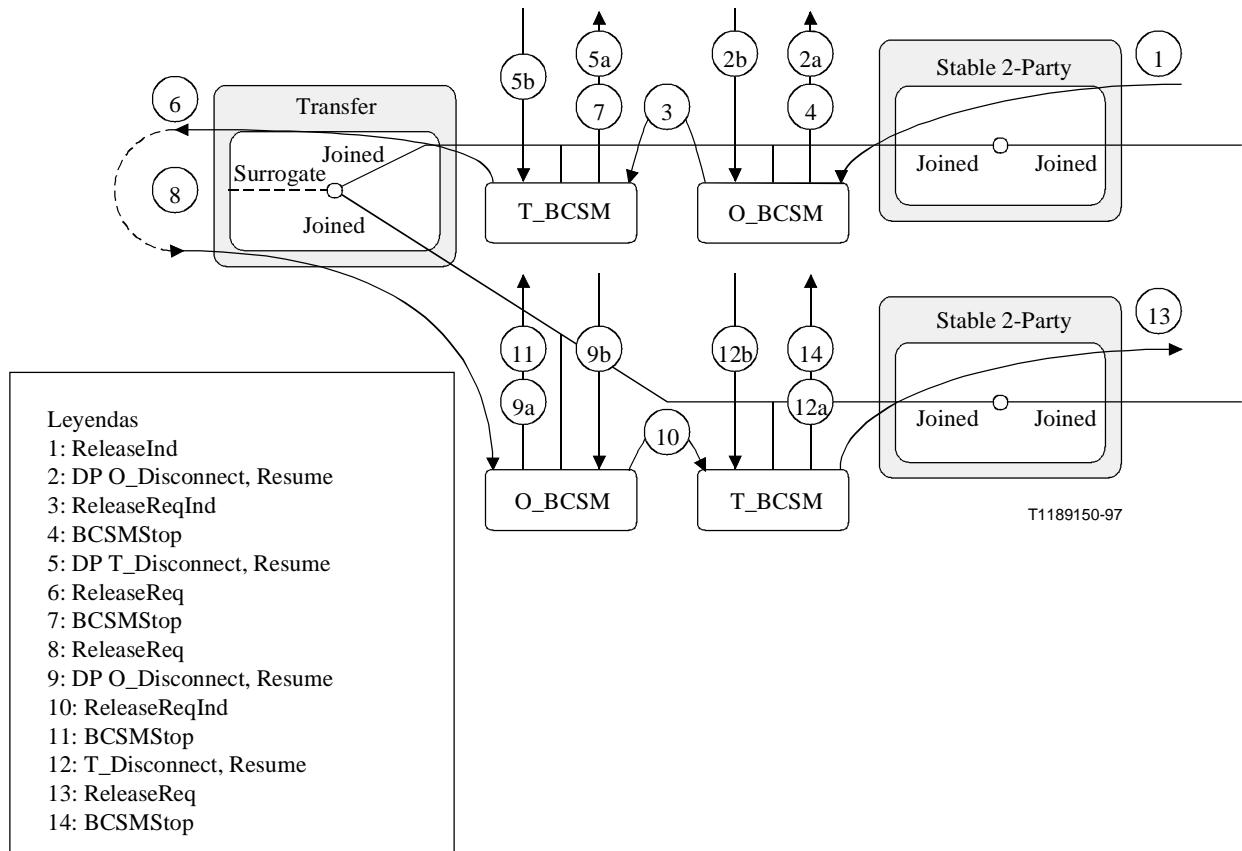
**Figura A.2-4/Q.1228 – Liberación de llamada/conexión cuando ambos segmentos de llamada están en "Stable\_2\_Party"**

En la figura A.2-5 se indica la liberación de una llamada/conexión para una llamada reenviada cuando los segmentos de la llamada están en el estado "bipartito estable" (Stable\_2\_Party) y en el estado "reenvío" (Forward).

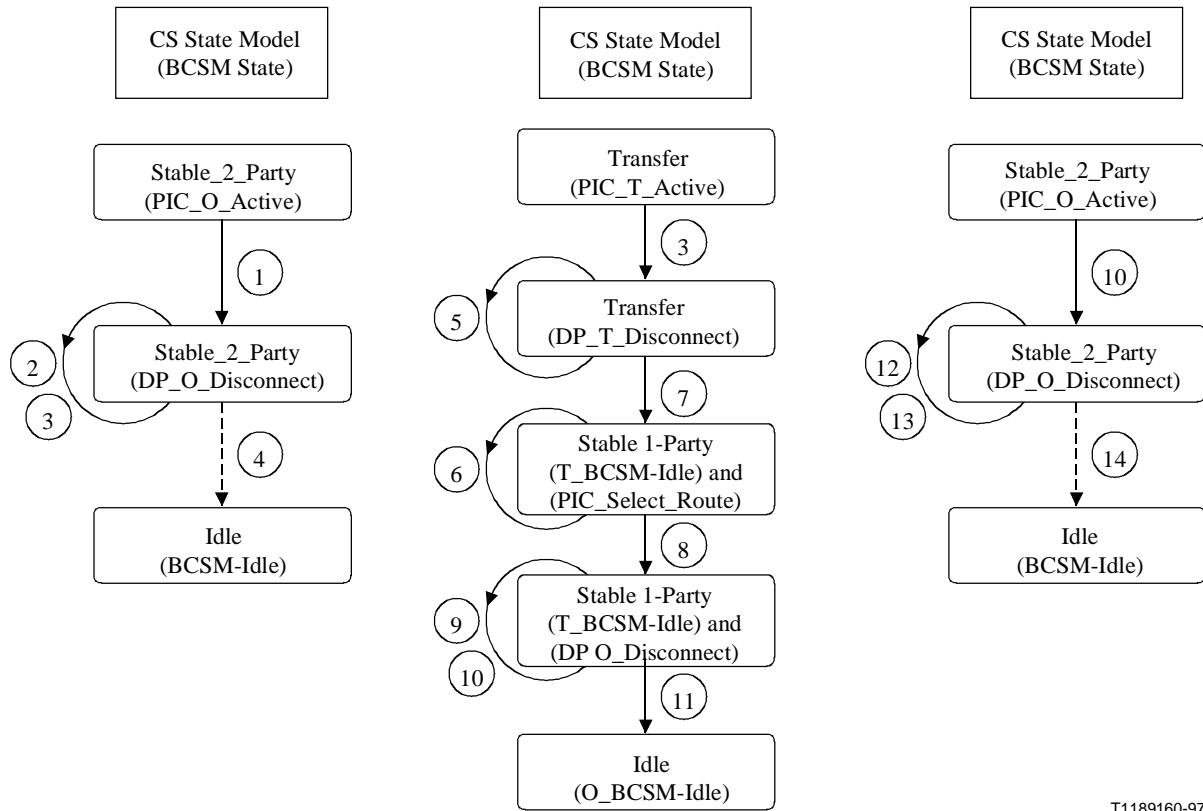


**Figura A.2-5/Q.1228 – Liberación de llamada/conexión cuando los segmentos de llamada están en el estado "Bipartito estable" (Stable\_2\_Party) y en el estado "Reenvío" (Forward)**

En las figuras A.2-6a y A.2-6b se indica la liberación de una llamada/conexión para una llamada reenviada cuando los segmentos de la llamada están en el estado "bipartito estable" (Stable\_2\_Party) y en el estado "transferencia" (Transfer).



**Figura A.2-6a/Q.1228 – Liberación de llamada/conexión cuando los segmentos de la llamada están en el estado "bipartito estable" (Stable\_2\_Party) y en el estado "transferencia" (Transfer) (flujo de información)**



T1189160-97

**Figura A.2-6b/Q.1228 – Liberación de llamada/conexión cuando los segmentos de la llamada están en el estado "bipartito estable" (Stable\_2\_Party) y en el estado "transferencia" (Transfer) (transiciones de estado)**

#### A.2.2.2.5 Operación desconexión de tramo

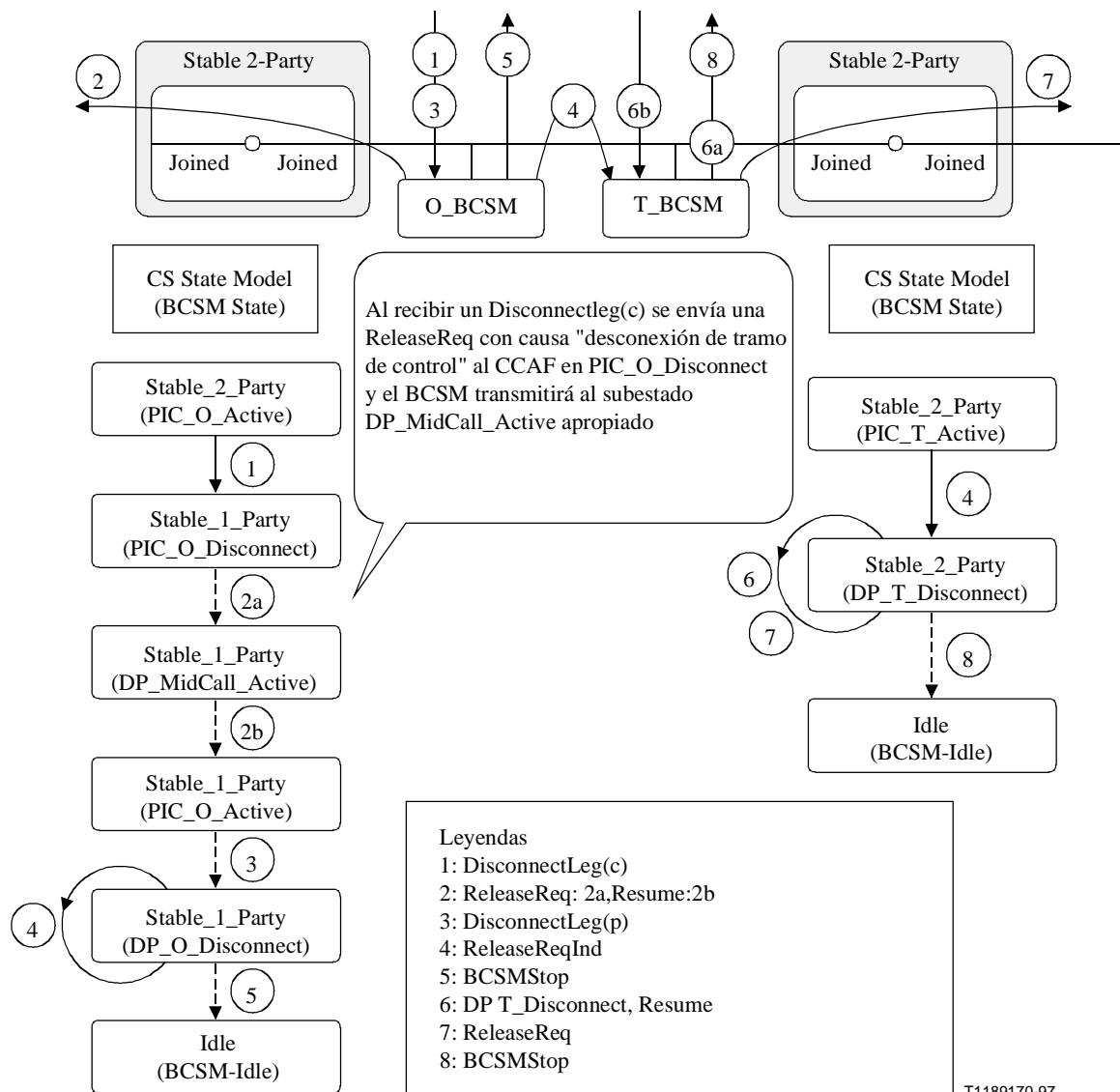
Una operación DisconnectLeg para un tramo controlador y un tramo pasivo liberará físicamente el tramo especificado desde el punto de conexión hacia el usuario distante. En la figura A.2-7 se indican los flujos de información que describen las secuencias de liberación como resultado de una operación DisconnectLeg para los tramos controlador (c) y pasivo de una llamada/conexión para una llamada normal en la que ambos segmentos de la llamada están en el estado "Stable\_2\_Party" (con el O\_BCSM y el T\_BCSM asociados en el estado activo).

Se indican los flujos de información para la recepción de las siguientes operaciones:

- Cuando se recibe una operación DisconnectLeg (desconexión de tramo) (para el tramo controlador) desde la lógica de servicio se producen las siguientes acciones:
  - se envía una señal ReleaseReq al CCAF con un valor de causa "desconexión del tramo controlador" en PIC\_O\_Disconnect (o PIC\_T\_Disconnect); y
  - el BCSM transitará al subestado MidCall\_Active PIC apropiado dependiendo de que un tramo pasivo siga o no asociado al tramo controlador.

NOTA – Cuando se reciba una operación Reconnect (reconexión) desde la lógica de servicio y un tramo pasivo distante siga asociado al tramo controlador, el BCSM volverá al estado BCSM anterior (O\_Active, O\_Suspended o T\_Active) después de la reanudación de la detección O\_Mid\_Call o T\_Mid\_Call apropiada (generada como resultado del comportamiento para la reconexión de una llamada desconectada).

- Cuando se recibe una operación DisconnectLeg (desconexión de tramo) (para el tramo pasivo) desde la lógica de servicio lógico se producen las siguientes acciones:
  - se envía una señal ReleaseReqInd al CCAF de BCSM distante con un valor de causa "desconexión del tramo pasivo" en DP\_O\_Disconnect (o DP\_T\_Disconnect); y
  - el BCSM transitará al subestado Disconnect PIC apropiado dependiendo de que un tramo de control siga o no asociado con el tramo pasivo.



**Figura A.2-7/Q.1228 – Desconexión de tramo de llamada/conexión cuando ambos segmentos de la llamada están en el estado "Bipartito estable" (Stable\_2\_Party)**

#### A.2.2.6 Interacciones de usuario durante el estado "supervisión" de la FSM de la SSF

Durante el estado "supervisión" de la FSM de la SSF será posible efectuar interacciones de usuario para enviar tonalidades y anuncios y visualizar información.

#### A.2.2.7 Numeración de tramos y reglas de detección dp asociadas

- i) La definición de los identificadores de tramo son las siguientes:
  - legID=1 y 2: indica la parte que estaba presente en el momento del "initialDp" o de una operación específica del DP, o la parte que fue creada con la operación "InitiateCallAttempt". Para un CS en estado establecimiento de origen (con un tramo controlador y un tramo pasivo conectado a un O\_BCSM) el legID controlador será "1" mientras que el tramo pasivo será 2. Para un CS en estado establecimiento de terminación (con un tramo controlador y un tramo pasivo conectado a un T\_BCSM), el legID controlador será "2" mientras que el tramo pasivo puede ser 1. Se señala que nunca puede utilizarse legID=0.
  - legID=3 a n: la SCF asignará los valores de tramo.
- ii) El principio de armado de puntos de detección es como sigue:
  - Todos los eventos a los cuales se aplica el filtrado (abandono, mitad de llamada y desconexión) y el control pueden ser armados como tramos pasivos según el sentido (cualquier sentido desde la parte que está conectada al tramo controlador o al tramo pasivo) en que tienen que ser capturados. Por ejemplo, el DP desconexión puede ser armado para el tramo controlador y el tramo pasivo. En ese caso, si se recibe del usuario una petición de liberación, ésta será detectada por el "DisconnectDP" armado para el tramo controlador, en tanto que una petición de liberación de las partes distantes se detectará mediante el armado del tramo pasivo "DisconnectDP" pertinente. Conviene señalar que por ahora no debe armarse ningún DP para el DP mitad de llamada y abandono para los tramos pasivos dado que sólo pueden recibirse peticiones del usuario conectado al tramo controlador.
  - Todos los eventos a los cuales no se aplica ningún principio de filtrado se armarán únicamente para el tramo pasivo.
- iii) Los siguientes valores por defecto para los eventos que deben armarse se aplican para la operación RequestReportBCSMEvent:
  - El valor por defecto para el armado será el tramo controlador para todos los eventos a los cuales se aplican los principios de filtrado (abandono, mitad de llamada y desconexión).
  - El valor por defecto para el armado será el tramo pasivo para todos los eventos a los cuales no se aplican los principios de filtrado.

#### A.2.2.8 Segmento de llamada y estados BCSM asociados para operaciones CPH

Se aplican los siguientes principios para el segmento de llamada y estados BCSM asociados para operaciones de tratamiento de los participantes en la llamada (CPH, *call party handling*):

- Si se recibe una operación CPH (SplitLeg, DisconnectLeg, MergeCallSegments, MoveCallSegment o MoveLeg) en el estado supervisión, las FSM para los segmentos de llamada que intervienen transitarán primero hacia el estado "en espera de instrucción" en tanto que los casos BCSM asociados dentro de los segmentos de llamada comprendidos se desplazarán a partir del punto en la llamada (PIC, *point in call*) O/T del correspondiente MidCallDP (punto de detección de mitad de llamada) O/T para tratar el rearmado EDP subsiguiente. Cabe destacar que cuando el BCSM está en un DP, permanecerá en éste después del procesamiento de la operación CPH. Esto implica que únicamente son posibles algunas transiciones PIC al MidCallDP, como se describe en las plantillas de las operaciones.
- La recepción de una operación CPH recibida y/o procesada en el estado "en espera de instrucción" no ocasionará el cambio del estado en espera de instrucción.

- Para las operaciones CPH que crean un nuevo BCSM para el tramo controlador (tramo trenzado), únicamente el nuevo BCSM será colocado en el DP-O/T-MidCall del estado activo. Los estados BCSM de los demás CS implicados deben desplazarse del PIC al estado DP correspondiente o permanecer en el estado de espera DP. La FSM de la SSF del CS asociado para el BCSM recientemente creado y los demás CS implicados se colocarán en el estado "en espera de instrucción" para que puedan rearmararse los EDP.

Todas las secuencias de la operación CPH finalizarán con una operación que para al estado en "supervisión" para cada tramo que interviene en el CS.

#### **A.2.2.2.9 Principios del comportamiento de la visión de la conexión "hacia adelante" y "transferencia"**

La visión de la conexión para los estados "hacia adelante" y "transferencia" se comportan de forma distinta cuando una operación es recibida desde el establecimiento de terminación (por ejemplo, servicio llamada hacia adelante) o de la parte 1 estable (por ejemplo, servicio conferencia con cita). Según la situación, los eventos de señalización recibidos desde una parte (un tramo) pueden ser retransmitidos a la otra parte. El método de procesamiento del evento de señalización recibido desde las partes está fuera del alcance del CS-2 de red inteligente.

#### **A.2.2.2.10 Convenios en la interfaz IBI**

Tiene que asignarse una bandera de llamada en la interfaz IBI para las señales que son bidireccionales, tales como ReleaseReqInd (indicación petición de liberación) y DataReqInd (indicación petición de datos). Para las señales que van de O\_BCSM a T\_BCSM (por ejemplo, SetupReqInd, indicación petición de establecimiento) o bien de T\_BCSM a O\_BCSM (por ejemplo, ProgressReqInd, indicación petición en moucha) no debe insertarse ninguna bandera de llamada.

Se aplican las siguientes convenciones para la bandera de llamada en la interfaz IBI:

- La bandera se coloca en <sender> (emisor) para las señales generadas desde O\_BCSM
- La bandera se coloca en <receiver> (receptor) para las señales generadas desde T\_BCSM

#### **A.2.2.3 Modelo de estados finitos**

La figura A.2-8 muestra las transiciones del subconjunto CS-1 de la RI del CS-2 de la RI para los estados de tipo de proceso de visión de conexión de segmentos de llamada.

La figura A.2-9 muestra las transiciones del CS-2 de la RI para los estados de tipo de proceso de visión de conexión de segmentos de llamada.

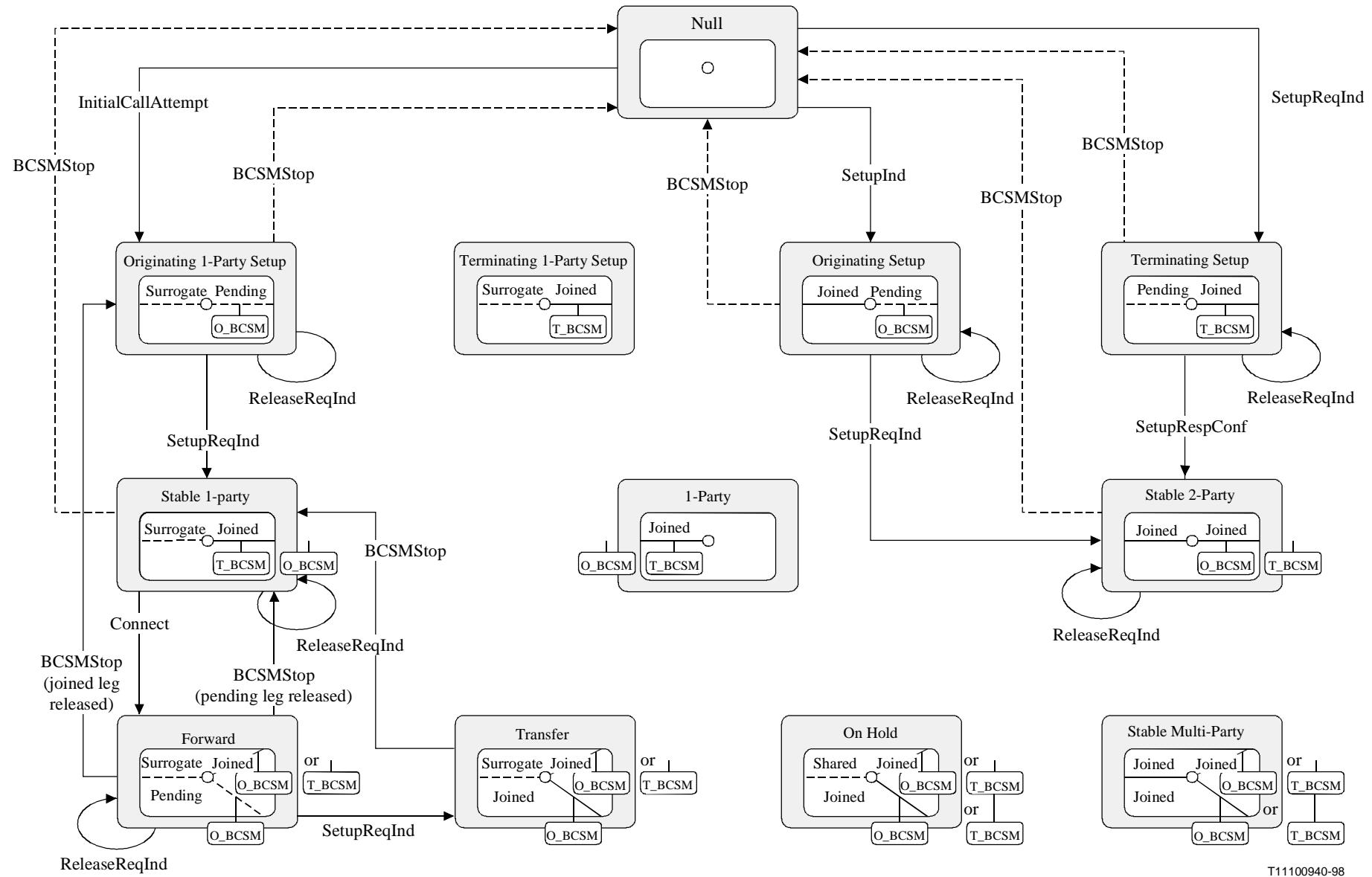


Figura A.2-8/Q.1228 – Modelo de estados finitos para la CSCV para el subconjunto CS-1 de la RI del CS-2 de la RI

T11100940-98

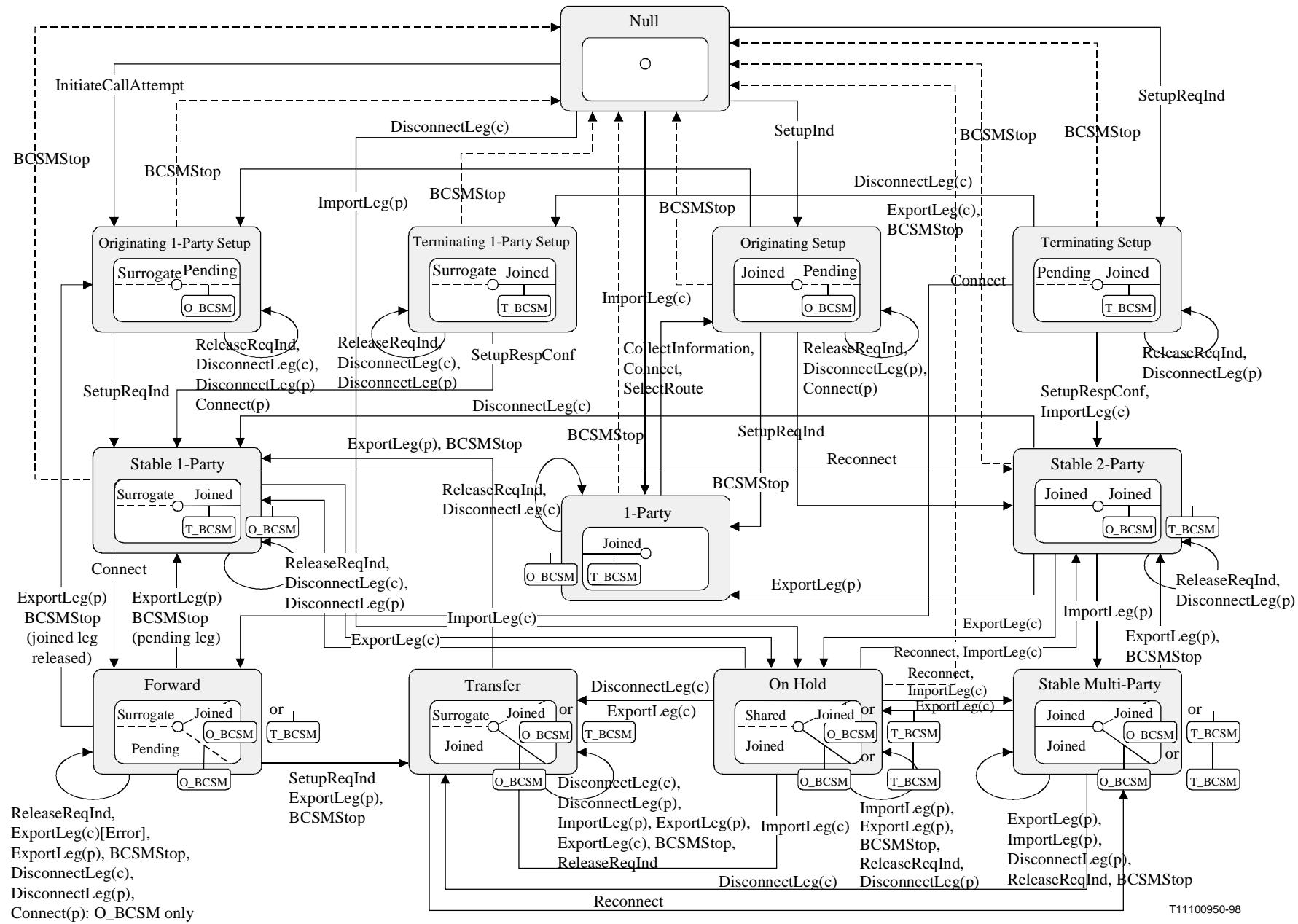


Figura A.2-9/A.1228 – Modelo de estados finitos para la CSCV del CS-2 de la RI

### **ANEXO A.3**

#### **Especificación en SDL de la SSF/CCF del CS-1**

Este anexo sólo está disponible en versión electrónica en los disquetes que van con este fascículo.

## **PARTE 6**

### **ANEXO A.4**

#### **Especificación en SDL de las extensiones a la SSF/CCF del CS-2**

Este anexo sólo está disponible en versión electrónica en los disquetes que van con este fascículo.

## **PARTE 7**

### **ANEXO A.5**

#### **Especificación en SDL de la SRF del CS-2**

Este anexo sólo está disponible en versión electrónica en los disquetes que van con este fascículo.

### **ANEXO A.6**

#### **Especificación en SDL de la SSF de asistencia/desasistencia del CS-2**

Este anexo sólo está disponible en versión electrónica en los disquetes que van con este fascículo.

### **ANEXO A.7**

#### **Especificación en SDL de la CUSF del CS-2**

Este anexo sólo está disponible en versión electrónica en los disquetes que van con este fascículo.

### **ANEXO A.8**

#### **Especificación en SDL de la SCF del CS-2**

Este anexo sólo está disponible en versión electrónica en los disquetes que van con este fascículo.

## APÉNDICE I

### Fuente ASN.1 expandida

Este apéndice contiene la fuente ASN.1 expandida para los tipos de datos comunes de la Recomendación Q.1228.

NOTA – Se efectuaron los cambios siguientes al módulo IN-CS2-datatypes para la generación de la fuente expandida (sólo para la compilación):

Componentes: relayedComponent se codificó como una cadena de caracteres (OCTET STRING)

ExtensionField: se codificó como una secuencia (SEQUENCE) vacía

TraceItem: el marcador de extensión (...) se eliminó del tipo SET

-- Expanded ASN.1 Module 'IN-CS2-datatypes'

-- Date: 97-09-23

**IN-CS2-datatypes { 0 0 17 1228 modules (0) in-cs2-datatypes (0) version1 (0) }**

#### **DEFINITIONS**

::=

#### **BEGIN**

**AccessCode ::= OCTET STRING (SIZE (??..??))**

**AccountNumber ::= NumericString (SIZE (1..151))**

**AChBillingChargingCharacteristics ::= OCTET STRING (SIZE (??..??))**

**ActionIndicator ::= ENUMERATED {  
    activate (1),  
    deactivate (2),  
    retrieve (3)}**

**ActionPerformed ::= ENUMERATED {  
    activated (1),  
    deactivated (2),  
    alreadyActive (3),  
    alreadyInactive (4),  
    isActive (5),  
    isInactive (6)}**

**ActivableServices ::= BIT STRING {  
    callingLineIdentificationPresentation (1),  
    callingLineIdentificationRestriction (2),  
    connectedLineIdentificationPresentation (3),  
    connectedLineIdentificationRestriction (4),  
    callForwardingOnNoReply (5),  
    callForwardingUnconditional (6),  
    callForwardingOnBusy (7),  
    callForwardingOnNotReachable (8),  
    reverseCharging (9),  
    adviceOfChargeOnStart (10),  
    adviceOfChargeAtEnd (11),  
    adviceOfChargeDuringCall (12),  
    timeDependentRouting (13),**

```
callingPartingDependentRouting (14),  
outgoingCallBarring (15),  
incomingCallBarring (16)}
```

AdditionalCallingPartyNumber ::= OCTET STRING (SIZE (??..??))

AlertingPattern ::= OCTET STRING (SIZE (3))

ApplicationTimer ::= INTEGER (0..2047)

AssistingSSPIPRoutingAddress ::= OCTET STRING (SIZE (??..??))

BackwardGVNS ::= OCTET STRING (SIZE (??..??))

```
BackwardServiceInteractionInd ::= SEQUENCE {  
    conferenceTreatmentIndicator [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,  
    callCompletionTreatmentIndicator [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL}
```

```
BCSMEvent ::= SEQUENCE {  
    eventTypeBCSM [0] IMPLICIT ENUMERATED {  
        origAttemptAuthorized (1),  
        collectedInfo (2),  
        analysedInformation (3),  
        routeSelectFailure (4),  
        oCalledPartyBusy (5),  
        oNoAnswer (6),  
        oAnswer (7),  
        oMidCall (8),  
        oDisconnect (9),  
        oAbandon (10),  
        termAttemptAuthorized (12),  
        tBusy (13),  
        tNoAnswer (14),  
        tAnswer (15),  
        tMidCall (16),  
        tDisconnect (17),  
        tAbandon (18),  
        oTermSeized (19),  
        oSuspended (20),  
        tSuspended (21),  
        origAttempt (22),  
        termAttempt (23),  
        oReAnswer (24),  
        tReAnswer (25),  
        facilitySelectedAndAvailable (26),  
        callAccepted (27)},  
    monitorMode [1] IMPLICIT ENUMERATED {  
        interrupted (0),  
        notifyAndContinue (1),  
        transparent (2)},  
    legID [2] CHOICE {  
        sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),  
        receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,  
    dpSpecificCriteria [30] CHOICE {  
        numberOfDigits [0] IMPLICIT INTEGER (1..255),  
        applicationTimer [1] IMPLICIT INTEGER (0..2047),  
        midCallControlInfo [2] IMPLICIT SEQUENCE SIZE (??..??) OF  
            . SEQUENCE {  
                . midCallInfoType [0] IMPLICIT SEQUENCE {  
                    . iNServiceControlCodeLow [0] IMPLICIT OCTET STRING (SIZE (??..??))},
```

- .
- *iNServiceControlCodeHigh* [1] IMPLICIT OCTET STRING (SIZE (??..??))  
    OPTIONAL},
- *midCallReportType* [1] IMPLICIT ENUMERATED {
- *inMonitoringState* (0),
- *inAnyState* (1)} DEFAULT *inMonitoringState* }} OPTIONAL}

**BCUSMEvent ::= SEQUENCE {**

- eventType [0] IMPLICIT ENUMERATED {**
- componentReceived* (127),
- associationReleaseRequested* (126)},
- monitorMode [1] IMPLICIT ENUMERATED {**
- interrupted* (0),
- notifyAndContinue* (1),
- transparent* (2)}

**BearerCapabilities ::= BIT STRING {**

- speech** (0),
- bc64kbits** (1),
- bc2x64kbits** (2),
- bc384kbits** (3),
- bc1536kbits** (4),
- bc1920kbits** (5),
- multirate** (6),
- restrictedDigitalInfo** (7),
- bc3-1khzAudio** (8),
- bc7khzAudio** (9),
- video** (10)}

**BearerCapability ::= CHOICE {**

- bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),**
- tmr [1] IMPLICIT OCTET STRING (SIZE (1))}**

**BothwayThroughConnectionInd ::= ENUMERATED {**

- bothwayPathRequired** (0),
- bothwayPathNotRequired** (1)}

**CallConditions ::= CHOICE {**

- userAbandon [0] IMPLICIT NULL,**
- callFailure [1] IMPLICIT OCTET STRING (SIZE (1)),**
- noReply [2] IMPLICIT INTEGER,**
- callRelease [3] IMPLICIT NULL,**
- ss-invocation [4] IMPLICIT ENUMERATED {**
- callingLineIdentificationRestriction* (1),
- connectedLineIdentificationRestriction* (2),
- callWaiting* (3),
- callHold* (4),
- reverseCharging* (5),
- explicitCallTransfer* (6),
- callCompletionOnBusySubscriber* (7)},
- creditLimitReached [5] IMPLICIT INTEGER,**
- callDuration [6] IMPLICIT INTEGER,**
- calledNumber [7] CHOICE {**
- initialMatch* [0] IMPLICIT OCTET STRING (SIZE (??..??)),
- totalMatch* [1] IMPLICIT OCTET STRING (SIZE (??..??))},
- answeredCall [8] IMPLICIT NULL}**

**CalledPartyBusinessGroupID ::= OCTET STRING**

**CalledPartyNumber ::= OCTET STRING (SIZE (??..??))**

**CalledPartySubaddress ::= OCTET STRING**

**CallIdentifier ::= INTEGER (1..2147483647)**  
**CallingPartyBusinessGroupID ::= OCTET STRING**  
**CallingPartyNumber ::= OCTET STRING (SIZE (??..??))**  
**CallingPartySubaddress ::= OCTET STRING**  
**CallingPartysCategory ::= OCTET STRING (SIZE (1))**  
**CallProcessingOperationCorrelationID ::= ENUMERATED {**  
    aLERTing (1),  
    sETUP (5),  
    cONNect (7),  
    dISConnect (69),  
    rELEASE (77),  
    rELeaseCOPplete (90),  
    fACility (98)}

**CallRecord ::= SEQUENCE {**  
    callDuration [0] IMPLICIT INTEGER (-2..86400),  
    callingPartyNumber [1] IMPLICIT OCTET STRING (SIZE (??..??)),  
    calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??))}

**CallResult ::= OCTET STRING (SIZE (??..??))**  
**CallSegmentID ::= INTEGER (1..???)**  
**initialCallSegment INTEGER ::= 1**

**CallUnrelatedDpSpecificCommonParameters ::= SEQUENCE {**  
    serviceAddressInformation [0] IMPLICIT SEQUENCE {  
        serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,  
        miscCallInfo [1] IMPLICIT SEQUENCE {  
            messageType [0] IMPLICIT ENUMERATED {  
                request (0),  
                notification (1)},  
            dpAssignment [1] IMPLICIT ENUMERATED {  
                individualLine (0),  
                groupBased (1),  
                officeBased (2)} OPTIONAL},  
            triggerType [2] IMPLICIT ENUMERATED {  
                featureActivation (0),  
                verticalServiceCode (1),  
                customizedAccess (2),  
                customizedIntercom (3),  
                emergencyService (12),  
                aFR . (13),  
                sharedIOTrunk (14),  
                offHookDelay (17),  
                channelSetupPRI (18),  
                tNoAnswer (25),  
                tBusy (26),  
                oCalledPartyBusy (27),  
                oNoAnswer (29),  
                originationAttemptAuthorized (30),  
                oAnswer (31),  
                oDisconnect (32),  
                termAttemptAuthorized (33),

```

        . tAnswer      (34),
        . tDisconnect (35)} OPTIONAL},
callingPartyNumber [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
locationNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
terminalType [3] IMPLICIT ENUMERATED {
    unknown      (0),
    dialPulse    (1),
    dtmf         (2),
    isdn         (3),
    isdnNoDtmf  (4),
    spare        (16)} DEFAULT isdn ,
extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {} OPTIONAL}

```

**Carrier** ::= OCTET STRING

**Cause** ::= OCTET STRING (SIZE (2..??))

**CauseValue** ::= OCTET STRING (SIZE (1))

**CGEncountered** ::= ENUMERATED {
 noCGencountered (0),
 manualCGencountered (1),
 scpOverload (2)}

**ChargeNumber** ::= OCTET STRING (SIZE (??..??))

**ChargingEvent** ::= SEQUENCE {
 eventTypeCharging [0] IMPLICIT OCTET STRING (SIZE (??..??)),
 monitorMode [1] IMPLICIT ENUMERATED {
 interrupted (0),
 notifyAndContinue (1),
 transparent (2)},
 legID [2] CHOICE {
 sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
 receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL}

**ChargingParameters** ::= SEQUENCE {
 unitsPerInterval [0] IMPLICIT INTEGER (0..??),
 timePerInterval [1] IMPLICIT INTEGER (0..??),
 scalingFactor [2] IMPLICIT INTEGER (0..??),
 initialUnitIncrement [3] IMPLICIT INTEGER (0..??) OPTIONAL,
 unitsPerDataInterval [4] IMPLICIT INTEGER (0..??) OPTIONAL,
 segmentsPerDataInterval [5] IMPLICIT INTEGER (0..??) OPTIONAL,
 initialTimeInterval [6] IMPLICIT INTEGER (0..??) OPTIONAL}

**CollectedDigits** ::= SEQUENCE {
 minimumNbOfDigits [0] IMPLICIT INTEGER (1..127) DEFAULT 1,
 maximumNbOfDigits [1] IMPLICIT INTEGER (1..127),
 endOfReplyDigit [2] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
 cancelDigit [3] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
 startDigit [4] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
 firstDigitTimeOut [5] IMPLICIT INTEGER (1..127) OPTIONAL,
 interDigitTimeOut [6] IMPLICIT INTEGER (1..127) OPTIONAL,
 errorTreatment [7] IMPLICIT ENUMERATED {
 reportErrorToScf (0),
 help (1),
 repeatPrompt (2)} DEFAULT reportErrorToScf ,
 interruptableAnnInd [8] IMPLICIT BOOLEAN DEFAULT TRUE,
 voiceInformation [9] IMPLICIT BOOLEAN DEFAULT FALSE,
 voiceBack [10] IMPLICIT BOOLEAN DEFAULT FALSE}

```

CollectedInfo ::= CHOICE {
    collectedDigits [0] IMPLICIT SEQUENCE {
        minimumNbOfDigits [0] IMPLICIT INTEGER (1..127) DEFAULT 1,
        maximumNbOfDigits [1] IMPLICIT INTEGER (1..127),
        endOfReplyDigit [2] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
        cancelDigit [3] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
        startDigit [4] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
        firstDigitTimeOut [5] IMPLICIT INTEGER (1..127) OPTIONAL,
        interDigitTimeOut [6] IMPLICIT INTEGER (1..127) OPTIONAL,
        errorTreatment [7] IMPLICIT ENUMERATED {
            . reportErrorToScf (0),
            . help (1),
            . repeatPrompt (2)} DEFAULT reportErrorToScf,
        interruptableAnnInd [8] IMPLICIT BOOLEAN DEFAULT TRUE,
        voiceInformation [9] IMPLICIT BOOLEAN DEFAULT FALSE,
        voiceBack [10] IMPLICIT BOOLEAN DEFAULT FALSE},
    iA5Information [1] IMPLICIT BOOLEAN}

```

```

Component ::= CHOICE {
    componentInfo [0] IMPLICIT OCTET STRING (SIZE (1..118)),
    relayedComponent [1] IMPLICIT OCTET STRING}

```

**ComponentCorrelationID ::= INTEGER**

```

ComponentType ::= ENUMERATED {
    any . (0),
    invoke (1),
    rResult (2),
    rError (3),
    rReject (4)}

```

```

ConnectedNumberTreatmentInd ::= ENUMERATED {
    noINImpact (0),
    presentationRestricted (1),
    presentCalledINNNumber (2)}

```

```

Constraints ::= SEQUENCE {
    maximumNumberOfDigits [1] IMPLICIT INTEGER (1..127),
    minimumNumberOfDigits [2] IMPLICIT INTEGER (1..127),
    typeOfRequestedInfo [3] IMPLICIT ENUMERATED {
        numericString (0),
        characterString (1),
        iA5String (2)} DEFAULT numericString ,
    numberOfAllowedRetries [4] IMPLICIT INTEGER (0..127) DEFAULT 0}

```

```

ControlConditionByCallParty ::= SEQUENCE {
    endOfMessageSendingDigit [0] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
    replayDigit [1] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL}

```

```

ControlType ::= ENUMERATED {
    sCPOverloaded (0),
    manuallyInitiated (1),
    destinationOverload (2)}

```

**CorrelationID ::= OCTET STRING (SIZE (??..??))**

```

CounterAndValue ::= SEQUENCE {
    counterID [0] IMPLICIT INTEGER (0..??),
    counterValue [1] IMPLICIT INTEGER (0..2147483647)}

```

```

CounterID ::= INTEGER (0..??)

CountersValue ::= SEQUENCE SIZE (0..100) OF
  SEQUENCE {
    counterID [0] IMPLICIT INTEGER (0..??),
    counterValue [1] IMPLICIT INTEGER (0..2147483647)}

Credit ::= CHOICE {
  currency   SEQUENCE {
    currency   PrintableString (SIZE (3)),
    amount     INTEGER (0..??)},
  units      INTEGER (0..65536)}

CreditUnit ::= INTEGER (0..65536)

CriticalityType ::= ENUMERATED {
  ignore      (0),
  abort       (1)}

CSAID      ::= INTEGER (1..??)

CurrencyID ::= PrintableString (SIZE (3))

CurrencyValue ::= SEQUENCE {
  currency   PrintableString (SIZE (3)),
  amount     INTEGER (0..??)}

CutAndPaste ::= INTEGER (0..22)

DateAndTime ::= OCTET STRING (SIZE (6))

DestinationRoutingAddress ::= SEQUENCE SIZE (1..3) OF
  OCTET STRING (SIZE (??..??))

Digits   ::= OCTET STRING (SIZE (??..??))

DisplayInformation ::= IA5String      (SIZE (??..??))

DpSpecificCommonParameters ::= SEQUENCE {
  serviceAddressInformation [0] IMPLICIT SEQUENCE {
    serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    miscCallInfo [1] IMPLICIT SEQUENCE {
      . messageType [0] IMPLICIT ENUMERATED {
        . request      (0),
        . notification (1)},
      . dpAssignment [1] IMPLICIT ENUMERATED {
        . individualLine (0),
        . groupBased   (1),
        . officeBased  (2)} OPTIONAL},
    triggerType [2] IMPLICIT ENUMERATED {
      . featureActivation (0),
      . verticalServiceCode (1),
      . customizedAccess (2),
      . customizedIntercom (3),
      . emergencyService (12),
      . aFR             (13),
      . sharedIOTrunk (14),
      . offHookDelay (17),
      . channelSetupPRI (18),
      . tNoAnswer (25),
      . tBusy          (26),
```

```

        .
        .   oCalledPartyBusy (27),
        .   oNoAnswer (29),
        .   originationAttemptAuthorized (30),
        .   oAnswer (31),
        .   oDisconnect (32),
        .   termAttemptAuthorized (33),
        .   tAnswer (34),
        .   tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
    tmr [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
    noCGEncountered (0),
    manualCGEncountered (1),
    scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {} OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceInteractionIndicators [15] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iNServiceCompatibilityIndication [16] IMPLICIT SEQUENCE SIZE (1..??) OF
    CHOICE {
        .
        .   agreements [0] IMPLICIT OBJECT IDENTIFIER,
        .   networkSpecific [1] IMPLICIT INTEGER (0..2147483647)} OPTIONAL,
serviceInteractionIndicatorsTwo [17] IMPLICIT SEQUENCE {
    forwardServiceInteractionInd [0] IMPLICIT SEQUENCE {
        .
        .   conferenceTreatmentIndicator [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
        .   callDiversionTreatmentIndicator [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
        .   callOfferingTreatmentIndicator [3] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL}
        OPTIONAL,
    backwardServiceInteractionInd [1] IMPLICIT SEQUENCE {
        .
        .   conferenceTreatmentIndicator [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
        .   callCompletionTreatmentIndicator [2] IMPLICIT OCTET STRING (SIZE (1))
        OPTIONAL} OPTIONAL,
    bothwayThroughConnectionInd [2] IMPLICIT ENUMERATED {
        .
        .   bothwayPathRequired (0),
        .   bothwayPathNotRequired (1)} OPTIONAL,
    suspendTimer [3] IMPLICIT INTEGER (0..120) OPTIONAL,
    connectedNumberTreatmentInd [4] IMPLICIT ENUMERATED {
        .
        .   noINImpact (0),
        .   presentationRestricted (1),
        .   presentCalledINNumber (2)} OPTIONAL,
    suppressCallDiversionNotification [5] IMPLICIT BOOLEAN OPTIONAL,
    suppressCallTransferNotification [6] IMPLICIT BOOLEAN OPTIONAL,

```

```

allowCdINoPresentationInd [7] IMPLICIT BOOLEAN OPTIONAL,
userDialogueDurationInd [8] IMPLICIT BOOLEAN DEFAULT TRUE,
... } OPTIONAL,
uSIServiceIndicator [18] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
uSIInformation [19] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
forwardGVNS [20] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
createdCallSegmentAssociation [21] IMPLICIT INTEGER (1..??) OPTIONAL,
... }

```

```

DpSpecificCriteria ::= CHOICE {
    number_of_digits [0] IMPLICIT INTEGER (1..255),
    application_timer [1] IMPLICIT INTEGER (0..2047),
    mid_call_control_info [2] IMPLICIT SEQUENCE SIZE (??..??) OF
        SEQUENCE {
            . mid_call_info_type [0] IMPLICIT SEQUENCE {
                . iNServiceControlCodeLow [0] IMPLICIT OCTET STRING (SIZE (??..??)),
                . iNServiceControlCodeHigh [1] IMPLICIT OCTET STRING (SIZE (??..??))
                    OPTIONAL},
                . mid_call_report_type [1] IMPLICIT ENUMERATED {
                    . inMonitoringState (0),
                    . inAnyState (1)} DEFAULT inMonitoringState }}}

```

**Duration** ::= INTEGER (-2..86400)

**ElementaryMessageID** ::= INTEGER (0..2147483647)

```

Entry ::= CHOICE {
    agreements [0] IMPLICIT OBJECT IDENTIFIER,
    network_specific [1] IMPLICIT INTEGER (0..2147483647)}

```

```

ErrorTreatment ::= ENUMERATED {
    reportErrorToScf (0),
    help (1),
    repeatPrompt (2)}

```

```

EventSpecificInformationBCSM ::= CHOICE {
    collected_info_specific_info [0] IMPLICIT SEQUENCE {
        called_party_number [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        ... },
    analysed_info_specific_info [1] IMPLICIT SEQUENCE {
        called_party_number [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        ... },
    route_select_failure_specific_info [2] IMPLICIT SEQUENCE {
        failure_cause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
        ... },
    oCalledPartyBusySpecificInfo [3] IMPLICIT SEQUENCE {
        busy_cause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
        ... },
    oNoAnswerSpecificInfo [4] IMPLICIT SEQUENCE {
        ... },
    oAnswerSpecificInfo [5] IMPLICIT SEQUENCE {
        backward_GVNS [0] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        ... },
    oMidCallSpecificInfo [6] IMPLICIT SEQUENCE {
        connect_time [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
        oMidCallInfo [1] IMPLICIT SEQUENCE {
            . iNServiceControlCode [0] IMPLICIT OCTET STRING (SIZE (??..??))} OPTIONAL,
        ... },
    ... }

```

```

oDisconnectSpecificInfo [7] IMPLICIT SEQUENCE {
    releaseCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    connectTime [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    ... },
tBusySpecificInfo [8] IMPLICIT SEQUENCE {
    busyCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    ... },
tNoAnswerSpecificInfo [9] IMPLICIT SEQUENCE {
    ... },
tAnswerSpecificInfo [10] IMPLICIT SEQUENCE {
    ... },
tMidCallSpecificInfo [11] IMPLICIT SEQUENCE {
    connectTime [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    tMidCallInfo [1] IMPLICIT SEQUENCE {
        .     iNServiceControlCode [0] IMPLICIT OCTET STRING (SIZE (??..??))} OPTIONAL,
    ... },
tDisconnectSpecificInfo [12] IMPLICIT SEQUENCE {
    releaseCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    connectTime [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    ... },
oTermSeizedSpecificInfo [13] IMPLICIT SEQUENCE {
    ... },
oSuspended [14] IMPLICIT SEQUENCE {
    ... },
tSuspended [15] IMPLICIT SEQUENCE {
    ... },
origAttemptAuthorized [16] IMPLICIT SEQUENCE {
    ... },
oReAnswer [17] IMPLICIT SEQUENCE {
    ... },
tReAnswer [18] IMPLICIT SEQUENCE {
    ... },
facilitySelectedAndAvailable [19] IMPLICIT SEQUENCE {
    ... },
callAccepted [20] IMPLICIT SEQUENCE {
    ... },
oAbandon [21] IMPLICIT SEQUENCE {
    abandonCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    ... },
tAbandon [22] IMPLICIT SEQUENCE {
    abandonCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
    ... }}EventSpecificInformationCharging ::= OCTET STRING (SIZE (??..??))

```

```

EventTypeBCSM ::= ENUMERATED {
    origAttemptAuthorized (1),
    collectedInfo (2),
    analysedInformation (3),
    routeSelectFailure (4),
    oCalledPartyBusy (5),
    oNoAnswer (6),
    oAnswer (7),
    oMidCall (8),
    oDisconnect (9),
    oAbandon (10),
    termAttemptAuthorized (12),
    tBusy (13),
    tNoAnswer (14),
    tAnswer (15),
    tMidCall (16),
    tDisconnect (17),
    tAbandon (18),
}

```

```

oTermSeized (19),
oSuspended (20),
tSuspended (21),
origAttempt (22),
termAttempt (23),
oReAnswer (24),
tReAnswer (25),
facilitySelectedAndAvailable (26),
callAccepted (27)}

```

```

EventTypeBCUSM ::= ENUMERATED {
    componentReceived (127),
    associationReleaseRequested (126)}

```

```
EventTypeCharging ::= OCTET STRING (SIZE (??..??))
```

```
ExtensionField ::= SEQUENCE {}
```

```

FacilityGroup ::= CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING}

```

```
FacilityGroupMember ::= INTEGER
```

```
FailureCause ::= OCTET STRING
```

```
FCIBillingChargingCharacteristics ::= OCTET STRING (SIZE (??..??))
```

```
FeatureCode ::= OCTET STRING (SIZE (??..??))
```

```

FeatureRequestIndicator ::= ENUMERATED {
    hold          (0),
    retrieve      (1),
    featureActivation (2),
    spare1        (3),
    sparen        (127)}

```

```

FilteredCallTreatment ::= SEQUENCE {
    sFBillingChargingCharacteristics [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    informationToSend [1] CHOICE {
        inbandInfo [0] IMPLICIT SEQUENCE {
            . messageID [0] CHOICE {
                . elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
                . text [1] IMPLICIT SEQUENCE {
                    . messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
                    . attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
                elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..?) OF
                    . INTEGER (0..2147483647),
                variableMessage [30] IMPLICIT SEQUENCE {
                    . elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
                    . variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
                        CHOICE {
                            . integer [0] IMPLICIT INTEGER (0..2147483647),
                            . number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
                            . time [2] IMPLICIT OCTET STRING (SIZE (2)),
                            . date [3] IMPLICIT OCTET STRING (SIZE (3)),
                            . price [4] IMPLICIT OCTET STRING (SIZE (4)))},
                . numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
                . duration [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
            }
        }
    }
}
```

```

.   interval [3] IMPLICIT INTEGER (0..32767) OPTIONAL},
tone   [1] IMPLICIT SEQUENCE {
.     toneID [0] IMPLICIT INTEGER (0..2147483647),
.     duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
displayInformation [2] IMPLICIT IA5String (SIZE (??..??))} OPTIONAL,
maximumNumberOfCounters [2] IMPLICIT INTEGER (1..100) OPTIONAL,
releaseCause [3] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL}

```

**FilteringCharacteristics ::= CHOICE {**

- interval [0] IMPLICIT INTEGER (1..32000),**
- numberOfCalls [1] IMPLICIT INTEGER (0..2147483647)}**

**FilteringCriteria ::= CHOICE {**

- dialledNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)),**
- callingLineID [1] IMPLICIT OCTET STRING (SIZE (??..??)),**
- serviceKey [2] IMPLICIT INTEGER (0..2147483647),**
- addressAndService [30] IMPLICIT SEQUENCE {**

  - calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),**
  - serviceKey [1] IMPLICIT INTEGER (0..2147483647),**
  - callingAddressValue [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,**
  - locationNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}}**

**FilteringTimeOut ::= CHOICE {**

- duration [0] IMPLICIT INTEGER (-2..86400),**
- stopTime [1] IMPLICIT OCTET STRING (SIZE (6))}**

**ForwardCallIndicators ::= OCTET STRING (SIZE (2))**

**ForwardGVNS ::= OCTET STRING (SIZE (??..??))**

**ForwardingCondition ::= ENUMERATED {**

- busy (0),**
- noanswer (1),**
- any . (2)}**

**ForwardServiceInteractionInd ::= SEQUENCE {**

- conferenceTreatmentIndicator [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,**
- callDiversionTreatmentIndicator [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,**
- callOfferingTreatmentIndicator [3] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL}**

**GapCriteria ::= CHOICE {**

- calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),**
- gapOnService [2] IMPLICIT SEQUENCE {**

  - serviceKey [0] IMPLICIT INTEGER (0..2147483647),**
  - dpCriteria [1] IMPLICIT ENUMERATED {**

    - . origAttemptAuthorized (1),**
    - . collectedInfo (2),**
    - . analysedInformation (3),**
    - . routeSelectFailure (4),**
    - . oCalledPartyBusy (5),**
    - . oNoAnswer (6),**
    - . oAnswer (7),**
    - . oMidCall (8),**
    - . oDisconnect (9),**
    - . oAbandon (10),**
    - . termAttemptAuthorized (12),**
    - . tBusy (13),**
    - . tNoAnswer (14),**
    - . tAnswer (15),**
    - . tMidCall (16),**
    - . tDisconnect (17),**
    - . tAbandon (18),**

```

.   oTermSeized (19),
.   oSuspended (20),
.   tSuspended (21),
.   origAttempt (22),
.   termAttempt (23),
.   oReAnswer (24),
.   tReAnswer (25),
.   facilitySelectedAndAvailable (26),
.   callAccepted (27} OPTIONAL},
gapAllInTraffic [3] IMPLICIT NULL,
calledAddressAndService [29] IMPLICIT SEQUENCE {
    calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    serviceKey [1] IMPLICIT INTEGER (0..2147483647)},
callingAddressAndService [30] IMPLICIT SEQUENCE {
    callingAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    serviceKey [1] IMPLICIT INTEGER (0..2147483647),
    locationNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}}

```

**GapOnService ::= SEQUENCE {**

- serviceKey [0] IMPLICIT INTEGER (0..2147483647),**
- dpCriteria [1] IMPLICIT ENUMERATED {**

  - origAttemptAuthorized (1),**
  - collectedInfo (2),**
  - analysedInformation (3),**
  - routeSelectFailure (4),**
  - oCalledPartyBusy (5),**
  - oNoAnswer (6),**
  - oAnswer (7),**
  - oMidCall (8),**
  - oDisconnect (9),**
  - oAbandon (10),**
  - termAttemptAuthorized (12),**
  - tBusy (13),**
  - tNoAnswer (14),**
  - tAnswer (15),**
  - tMidCall (16),**
  - tDisconnect (17),**
  - tAbandon (18),**
  - oTermSeized (19),**
  - oSuspended (20),**
  - tSuspended (21),**
  - origAttempt (22),**
  - termAttempt (23),**
  - oReAnswer (24),**
  - tReAnswer (25),**
  - facilitySelectedAndAvailable (26),**
  - callAccepted (27)} OPTIONAL}**

**GapIndicators ::= SEQUENCE {**

- duration [0] IMPLICIT INTEGER (-2..86400),**
- gapInterval [1] IMPLICIT INTEGER (-1..60000)}**

**GapTreatment ::= CHOICE {**

- informationToSend [0] CHOICE {**
- inbandInfo [0] IMPLICIT SEQUENCE {**
- messageID [0] CHOICE {**
- elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),**
- text [1] IMPLICIT SEQUENCE {**
- . messageContent [0] IMPLICIT IA5String (SIZE (??..??)),**
- . attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},**
- elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF**

```

.     INTEGER (0..2147483647),
variableMessage [30] IMPLICIT SEQUENCE {
.     elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
.     variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
.         CHOICE {
.             .     integer      [0] IMPLICIT INTEGER (0..2147483647),
.             .     number       [1] IMPLICIT OCTET STRING (SIZE (??..??)),
.             .     time        [2] IMPLICIT OCTET STRING (SIZE (2)),
.             .     date        [3] IMPLICIT OCTET STRING (SIZE (3)),
.             .     price       [4] IMPLICIT OCTET STRING (SIZE (4))}},
.     numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
.     duration      [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
.     interval      [3] IMPLICIT INTEGER (0..32767) OPTIONAL},
tone   [1] IMPLICIT SEQUENCE {
.     toneID      [0] IMPLICIT INTEGER (0..2147483647),
.     duration      [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
displayInformation [2] IMPLICIT IA5String (SIZE (??..??))),
releaseCause [1] IMPLICIT OCTET STRING (SIZE (2..??)),
both    [2] IMPLICIT SEQUENCE {
informationToSend [0] CHOICE {
inbandInfo [0] IMPLICIT SEQUENCE {
messageID [0] CHOICE {
.     elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
.     text          [1] IMPLICIT SEQUENCE {
.         messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
.         attributes    [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..?) OF
.         INTEGER (0..2147483647),
variableMessage [30] IMPLICIT SEQUENCE {
.         elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
.         variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
.             CHOICE {
.                 .     integer      [0] IMPLICIT INTEGER (0..2147483647),
.                 .     number       [1] IMPLICIT OCTET STRING (SIZE (??..??)),
.                 .     time        [2] IMPLICIT OCTET STRING (SIZE (2)),
.                 .     date        [3] IMPLICIT OCTET STRING (SIZE (3)),
.                 .     price       [4] IMPLICIT OCTET STRING (SIZE (4))}},
.         numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
duration      [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
interval      [3] IMPLICIT INTEGER (0..32767) OPTIONAL},
tone   [1] IMPLICIT SEQUENCE {
toneID      [0] IMPLICIT INTEGER (0..2147483647),
duration      [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
displayInformation [2] IMPLICIT IA5String (SIZE (??..??))),
releaseCause [1] IMPLICIT OCTET STRING (SIZE (2..??)))}

```

**GenericName** ::= OCTET STRING (SIZE (??..??))

**GenericNumber** ::= OCTET STRING (SIZE (??..??))

**GenericNumbers** ::= SET SIZE (1..?) OF  
OCTET STRING (SIZE (??..??))

**HighLayerCompatibilities** ::= BIT STRING {  
telephony (0),  
facsimileGroup2-3 (1),  
facsimileGroup4classeI (2),  
teletexMixedMode (3),  
teletexProcessableMode (4),  
teletexBasicMode (5),  
syntaxBasedVideotex (6),

```

internationalVideotex (7),
telexService (8),
messageHandlingSystem (9),
osiApplication (10),
audioVisual (11)}

```

**HighLayerCompatibility ::= OCTET STRING (SIZE (2))**

**HoldCause ::= OCTET STRING**

```

InbandInfo ::= SEQUENCE {
    messageID [0] CHOICE {
        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
        text [1] IMPLICIT SEQUENCE {
            . messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
            . attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
        elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF
            . INTEGER (0..2147483647),
        variableMessage [30] IMPLICIT SEQUENCE {
            . elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
            . variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
                CHOICE {
                    . integer [0] IMPLICIT INTEGER (0..2147483647),
                    . number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
                    . time [2] IMPLICIT OCTET STRING (SIZE (2)),
                    . date [3] IMPLICIT OCTET STRING (SIZE (3)),
                    . price [4] IMPLICIT OCTET STRING (SIZE (4)))}},
    numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
    duration [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
    interval [3] IMPLICIT INTEGER (0..32767) OPTIONAL}

```

```

InformationToRecord ::= SEQUENCE {
    messageID [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    messageDeletionTimeOut [1] IMPLICIT INTEGER (1..3600) OPTIONAL,
    timeToRecord [3] IMPLICIT INTEGER (0..??) OPTIONAL,
    controlDigits [4] IMPLICIT SEQUENCE {
        endOfRecordingDigit [0] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
        cancelDigit [1] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
        replayDigit [2] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
        restartRecordingDigit [3] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
        restartAllowed [4] IMPLICIT BOOLEAN DEFAULT FALSE,
        replayAllowed [5] IMPLICIT BOOLEAN DEFAULT FALSE}}

```

```

InformationToSend ::= CHOICE {
    inbandInfo [0] IMPLICIT SEQUENCE {
        messageID [0] CHOICE {
            . elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
            . text [1] IMPLICIT SEQUENCE {
                . messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
                . attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
            elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF
                . INTEGER (0..2147483647),
            variableMessage [30] IMPLICIT SEQUENCE {
                . elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
                . variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
                    CHOICE {
                        . integer [0] IMPLICIT INTEGER (0..2147483647),
                        . number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
                        . time [2] IMPLICIT OCTET STRING (SIZE (2)),
                        . date [3] IMPLICIT OCTET STRING (SIZE (3)),
                        . price [4] IMPLICIT OCTET STRING (SIZE (4)))}}}}

```

```

numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
duration      [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
interval      [3] IMPLICIT INTEGER (0..32767) OPTIONAL},
tone          [1] IMPLICIT SEQUENCE {
    toneID        [0] IMPLICIT INTEGER (0..2147483647),
    duration      [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
displayInformation [2] IMPLICIT IA5String (SIZE (??..??))}

InfoToSend ::= CHOICE {
    messageID [0] CHOICE {
        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
        text           [1] IMPLICIT SEQUENCE {
            .   messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
            .   attributes     [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
        elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF
            .   INTEGER (0..2147483647),
        variableMessage [30] IMPLICIT SEQUENCE {
            .   elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
            .   variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
                CHOICE {
                    .   .   integer      [0] IMPLICIT INTEGER (0..2147483647),
                    .   .   number       [1] IMPLICIT OCTET STRING (SIZE (??..??)),
                    .   .   time         [2] IMPLICIT OCTET STRING (SIZE (2)),
                    .   .   date         [3] IMPLICIT OCTET STRING (SIZE (3)),
                    .   .   price        [4] IMPLICIT OCTET STRING (SIZE (4))}},
        toneId [1] CHOICE {
            local        [0] IMPLICIT INTEGER (0..2147483647),
            global       [1] IMPLICIT OBJECT IDENTIFIER},
        displayInformation [2] IMPLICIT IA5String (SIZE (??..??))}

InfoType ::= ENUMERATED {
    numericString (0),
    characterString (1),
    iA5String     (2)}

INServiceCompatibilityIndication ::= SEQUENCE SIZE (1..??) OF
CHOICE {
    agreements [0] IMPLICIT OBJECT IDENTIFIER,
    networkSpecific [1] IMPLICIT INTEGER (0..2147483647)}

INServiceCompatibilityResponse ::= CHOICE {
    agreements [0] IMPLICIT OBJECT IDENTIFIER,
    networkSpecific [1] IMPLICIT INTEGER (0..2147483647)}

Integer4 ::= INTEGER (0..2147483647)

InteractionStrategy ::= ENUMERATED {
    stopOnError (1),
    bestEffort  (2)}

Interval ::= INTEGER (-1..60000)

InvokableService ::= ENUMERATED {
    callingLineIdentificationRestriction (1),
    connectedLineIdentificationRestriction (2),
    callWaiting (3),
    callHold    (4),
    reverseCharging (5),
    explicitCallTransfer (6),
    callCompletionOnBusySubscriber (7)}

```

**InvokeID** ::= INTEGER (-128..127)  
**IPAvailable** ::= OCTET STRING (SIZE (??..??))  
**IPRoutingAddress** ::= OCTET STRING (SIZE (??..??))  
**IPSSPCapabilities** ::= OCTET STRING (SIZE (??..??))  
**ISDNAccessRelatedInformation** ::= OCTET STRING  
**Language** ::= PrintableString (SIZE (3))  
**LegID** ::= CHOICE {  
 sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),  
 receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))}  
**LegType** ::= OCTET STRING (SIZE (1))  
**leg1** OCTET STRING (SIZE (1)) ::= '01'H  
**leg2** OCTET STRING (SIZE (1)) ::= '02'H  
**LocationNumber** ::= OCTET STRING (SIZE (??..??))  
**MailBoxID** ::= OCTET STRING (SIZE (??..??))  
**MaximumNumberOfCounters** ::= INTEGER (1..100)  
**Media** ::= ENUMERATED {  
 voiceMail (0),  
 faxGroup3 (1),  
 faxGroup4 (2)}  
**Message** ::= ENUMERATED {  
 rERelease (77),  
 rEReleaseCOMplete (90),  
 fACility (98)}  
**MessageID** ::= CHOICE {  
 elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),  
 text [1] IMPLICIT SEQUENCE {  
 messageContent [0] IMPLICIT IA5String (SIZE (??..??)),  
 attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},  
 elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF  
 INTEGER (0..2147483647),  
 variableMessage [30] IMPLICIT SEQUENCE {  
 elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),  
 variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF  
 . CHOICE {  
 . integer [0] IMPLICIT INTEGER (0..2147483647),  
 . number [1] IMPLICIT OCTET STRING (SIZE (??..??)),  
 . time [2] IMPLICIT OCTET STRING (SIZE (2)),  
 . date [3] IMPLICIT OCTET STRING (SIZE (3)),  
 . price [4] IMPLICIT OCTET STRING (SIZE (4))}}}  
**MidCallControlInfo** ::= SEQUENCE SIZE (??..??) OF  
 SEQUENCE {  
 midCallInfoType [0] IMPLICIT SEQUENCE {  
 . iNServiceControlCodeLow [0] IMPLICIT OCTET STRING (SIZE (??..??)),  
 . iNServiceControlCodeHigh [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},  
 }

```

midCallReportType [1] IMPLICIT ENUMERATED {
    . inMonitoringState (0),
    . inAnyState (1)} DEFAULT inMonitoringState }

MidCallInfo ::= SEQUENCE {
    iNServiceControlCode [0] IMPLICIT OCTET STRING (SIZE (??..??))}

MidCallInfoType ::= SEQUENCE {
    iNServiceControlCodeLow [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    iNServiceControlCodeHigh [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}

MiscCallInfo ::= SEQUENCE {
    messageType [0] IMPLICIT ENUMERATED {
        request (0),
        notification (1)},
    dpAssignment [1] IMPLICIT ENUMERATED {
        individualLine (0),
        groupBased (1),
        officeBased (2)} OPTIONAL}

MonitorMode ::= ENUMERATED {
    interrupted (0),
    notifyAndContinue (1),
    transparent (2)}

Notification ::= ENUMERATED {
    userAbandon (0),
    callFailure (1),
    noReply (2),
    callRelease (3),
    ssInvocation (4),
    creditLimitReached (5),
    callDuration (6),
    calledNumber (7),
    answeredCall (8)}

NotificationInformation ::= CHOICE {
    userAbandonSpecificInfo [0] IMPLICIT SEQUENCE {
        ... },
    callFailureSpecificInfo [1] IMPLICIT SEQUENCE {
        failureCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
        ... },
    noReplySpecificInfo [2] IMPLICIT SEQUENCE {
        ... },
    callReleaseSpecificInfo [3] IMPLICIT SEQUENCE {
        releaseCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
        timeStamp [1] IMPLICIT OCTET STRING (SIZE (6)) OPTIONAL,
        ... },
    ssInvocationSpecificInfo [4] IMPLICIT SEQUENCE {
        invokedService [0] IMPLICIT ENUMERATED {
            . callingLineIdentificationRestriction (1),
            . connectedLineIdentificationRestriction (2),
            . callWaiting (3),
            . callHold (4),
            . reverseCharging (5),
            . explicitCallTransfer (6),
            . callCompletionOnBusySubscriber (7)},
        ... },
    creditLimitReachedSpecificInfo [5] IMPLICIT SEQUENCE {
        timeStamp [0] IMPLICIT OCTET STRING (SIZE (6)) OPTIONAL,
        ... },
    ...
}

```

```

callDurationSpecificInfo [6] IMPLICIT SEQUENCE {
    timeStamp [0] IMPLICIT OCTET STRING (SIZE (6)) OPTIONAL,
    ...
},
calledNumberSpecificInfo [7] IMPLICIT SEQUENCE {
    calledNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    ...
},
answeredCallSpecificInfo [8] IMPLICIT SEQUENCE {
    timeStamp [0] IMPLICIT OCTET STRING (SIZE (6)) OPTIONAL,
    ...
}

```

**NumberingPlan** ::= OCTET STRING (SIZE (1))

**NumberMatch** ::= CHOICE {
 **initialMatch** [0] IMPLICIT OCTET STRING (SIZE (??..??)),
 **totalMatch** [1] IMPLICIT OCTET STRING (SIZE (??..??))}

**NumberOfDigits** ::= INTEGER (1..255)

**OperationCode** ::= CHOICE {
 **globalCode** OBJECT IDENTIFIER,
 **local** INTEGER}

**OriginalCalledPartyID** ::= OCTET STRING (SIZE (??..??))

**ProfileIdentifier** ::= CHOICE {
 **access** [0] IMPLICIT OCTET STRING (SIZE (??..??)),
 **group** [1] CHOICE {
 **trunkGroupID** [0] IMPLICIT INTEGER,
 **privateFacilityID** [1] IMPLICIT INTEGER,
 **huntGroup** [2] IMPLICIT OCTET STRING,
 **routeIndex** [3] IMPLICIT OCTET STRING}}

**Reason** ::= OCTET STRING (SIZE (??..??))

**ReceivedInformation** ::= SEQUENCE SIZE (??..??) OF  
**IA5String**

**ReceivedStatus** ::= ENUMERATED {
 **messageComplete** (0),
 **messageInterrupted** (1),
 **messageTimeOut** (2)}

**RecordedMessageID** ::= INTEGER (0..2147483647)

**RedirectingPartyID** ::= OCTET STRING (SIZE (??..??))

**RedirectionInformation** ::= OCTET STRING (SIZE (2))

**RegistrarIdentifier** ::= OCTET STRING

**ReportCondition** ::= ENUMERATED {
 **statusReport** (0),
 **timerExpired** (1),
 **canceled** (2)}

**RequestedInformationList** ::= SEQUENCE SIZE (1..5) OF  
SEQUENCE {
 **requestedInformationType** [0] IMPLICIT ENUMERATED {
 .
 .
 .
 **callAttemptElapsedTIme** (0),
 **callStopTime** (1),
 **callConnectedElapsedTIme** (2),
 }
}

```

.   calledAddress (3),
.   releaseCause (30)},
requestedInformationValue [1] CHOICE {
.   callAttemptElapsed TValue [0] IMPLICIT INTEGER (0..255),
.   callStopTimeValue [1] IMPLICIT OCTET STRING (SIZE (6)),
.   callConnectedElapsed TValue [2] IMPLICIT INTEGER (0..2147483647),
.   calledAddressValue [3] IMPLICIT OCTET STRING (SIZE (??..??)),
.   releaseCauseValue [30] IMPLICIT OCTET STRING (SIZE (2..??))}}

```

**RequestedInformationTypeList ::= SEQUENCE SIZE (1..5) OF**

```

ENUMERATED {
    callAttemptElapsed (0),
    callStopTime (1),
    callConnectedElapsed (2),
    calledAddress (3),
    releaseCause (30)}

```

**RequestedInformation ::= SEQUENCE {**

```

requestedInformationType [0] IMPLICIT ENUMERATED {
    callAttemptElapsed (0),
    callStopTime (1),
    callConnectedElapsed (2),
    calledAddress (3),
    releaseCause (30)},
requestedInformationValue [1] CHOICE {
    callAttemptElapsed TValue [0] IMPLICIT INTEGER (0..255),
    callStopTimeValue [1] IMPLICIT OCTET STRING (SIZE (6)),
    callConnectedElapsed TValue [2] IMPLICIT INTEGER (0..2147483647),
    calledAddressValue [3] IMPLICIT OCTET STRING (SIZE (??..??)),
    releaseCauseValue [30] IMPLICIT OCTET STRING (SIZE (2..??))}}

```

**RequestedInformationType ::= ENUMERATED {**

```

    callAttemptElapsed (0),
    callStopTime (1),
    callConnectedElapsed (2),
    calledAddress (3),
    releaseCause (30)}

```

**RequestedInformationValue ::= CHOICE {**

```

    callAttemptElapsed TValue [0] IMPLICIT INTEGER (0..255),
    callStopTimeValue [1] IMPLICIT OCTET STRING (SIZE (6)),
    callConnectedElapsed TValue [2] IMPLICIT INTEGER (0..2147483647),
    calledAddressValue [3] IMPLICIT OCTET STRING (SIZE (??..??)),
    releaseCauseValue [30] IMPLICIT OCTET STRING (SIZE (2..??))}

```

**RequestedNotifications ::= SET OF**

```

CHOICE {
    userAbandon [0] IMPLICIT NULL,
    callFailure [1] IMPLICIT OCTET STRING (SIZE (1)),
    noReply [2] IMPLICIT INTEGER,
    callRelease [3] IMPLICIT NULL,
    ss-invocation [4] IMPLICIT ENUMERATED {
        . callingLineIdentificationRestriction (1),
        . connectedLineIdentificationRestriction (2),
        . callWaiting (3),
        . callHold (4),
        . reverseCharging (5),
        . explicitCallTransfer (6),
        . callCompletionOnBusySubscriber (7)},
    creditLimitReached [5] IMPLICIT INTEGER,
    callDuration [6] IMPLICIT INTEGER,

```

```

calledNumber [7] CHOICE {
    . initialMatch [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    . totalMatch [1] IMPLICIT OCTET STRING (SIZE (??..??))},
answeredCall [8] IMPLICIT NULL}

```

**RequestedType** ::= INTEGER (0..127)

```

RequestedUTSI ::= SEQUENCE {
    uSIServiceIndicator [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    uSImonitorMode [1] IMPLICIT ENUMERATED {
        monitoringActive (0),
        monitoringInactive (1)},
    legID [2] CHOICE {
        sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
        receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} DEFAULT sendingSideID : '01'H}

```

**RequestedUTSIList** ::= SEQUENCE SIZE (??..??) OF

```

SEQUENCE {
    uSIServiceIndicator [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    uSImonitorMode [1] IMPLICIT ENUMERATED {
        . monitoringActive (0),
        . monitoringInactive (1)},
    legID [2] CHOICE {
        . sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
        . receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} DEFAULT sendingSideID :
        '01'H}

```

```

ResourceID ::= CHOICE {
    lineID [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    facilityGroupID [1] CHOICE {
        trunkGroupID [0] IMPLICIT INTEGER,
        privateFacilityID [1] IMPLICIT INTEGER,
        huntGroup [2] IMPLICIT OCTET STRING,
        routeIndex [3] IMPLICIT OCTET STRING},
    facilityGroupMemberID [2] IMPLICIT INTEGER,
    trunkGroupID [3] IMPLICIT INTEGER}

```

```

ResourceStatus ::= ENUMERATED {
    busy (0),
    idle (1)}

```

```

ResponseCondition ::= ENUMERATED {
    intermediateResponse (0),
    lastResponse (1)}

```

**RouteList** ::= SEQUENCE SIZE (1..3) OF  
OCTET STRING (SIZE (??..??))

```

RoutingAddress ::= CHOICE {
    routingProhibited [0] IMPLICIT NULL,
    destinationRoutingAddress [1] IMPLICIT SEQUENCE SIZE (1..3) OF
        OCTET STRING (SIZE (??..??))}

```

**ScfAddress** ::= OCTET STRING (SIZE (??..??))

**ScfID** ::= OCTET STRING (SIZE (??..??))

**SCIBillingChargingCharacteristics** ::= OCTET STRING (SIZE (??..??))

```

ServiceAddressInformation ::= SEQUENCE {
    serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    miscCallInfo [1] IMPLICIT SEQUENCE {
        messageType [0] IMPLICIT ENUMERATED {
            . request (0),
            . notification (1)},
        dpAssignment [1] IMPLICIT ENUMERATED {
            . individualLine (0),
            . groupBased (1),
            . officeBased (2)} OPTIONAL},
    triggerType [2] IMPLICIT ENUMERATED {
        featureActivation (0),
        verticalServiceCode (1),
        customizedAccess (2),
        customizedIntercom (3),
        emergencyService (12),
        aFR . (13),
        sharedIOTrunk (14),
        offHookDelay (17),
        channelSetupPRI (18),
        tNoAnswer (25),
        tBusy (26),
        oCalledPartyBusy (27),
        oNoAnswer (29),
        originationAttemptAuthorized (30),
        oAnswer (31),
        oDisconnect (32),
        termAttemptAuthorized (33),
        tAnswer (34),
        tDisconnect (35)} OPTIONAL}

```

ServiceInteractionIndicators ::= OCTET STRING (SIZE (??..??))

```

ServiceInteractionIndicatorsTwo ::= SEQUENCE {
    forwardServiceInteractionInd [0] IMPLICIT SEQUENCE {
        conferenceTreatmentIndicator [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
        callDiversionTreatmentIndicator [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
        callOfferingTreatmentIndicator [3] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL}
    OPTIONAL,
    backwardServiceInteractionInd [1] IMPLICIT SEQUENCE {
        conferenceTreatmentIndicator [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
        callCompletionTreatmentIndicator [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL}
    OPTIONAL,
    bothwayThroughConnectionInd [2] IMPLICIT ENUMERATED {
        bothwayPathRequired (0),
        bothwayPathNotRequired (1)} OPTIONAL,
    suspendTimer [3] IMPLICIT INTEGER (0..120) OPTIONAL,
    connectedNumberTreatmentInd [4] IMPLICIT ENUMERATED {
        noINImpact (0),
        presentationRestricted (1),
        presentCalledINNumber (2)} OPTIONAL,
    suppressCallDiversionNotification [5] IMPLICIT BOOLEAN OPTIONAL,
    suppressCallTransferNotification [6] IMPLICIT BOOLEAN OPTIONAL,
    allowCdINNoPresentationInd [7] IMPLICIT BOOLEAN OPTIONAL,
    userDialogueDurationInd [8] IMPLICIT BOOLEAN DEFAULT TRUE,
    ... }

```

ServiceKey ::= INTEGER (0..2147483647)

ServiceProfileIdentifier ::= OCTET STRING

ServingAreaID ::= OCTET STRING (SIZE (??..??))

**SFBillingChargingCharacteristics ::= OCTET STRING (SIZE (??..??))**

**SubscriberId ::= OCTET STRING (SIZE (??..??))**

**SupplementaryServices ::= BIT STRING {**

- callingLineIdentificationPresentation (1),**
- callingLineIdentificationRestriction (2),**
- connectedLineIdentificationPresentation (3),**
- connectedLineIdentificationRestriction (4),**
- callForwardingOnNoReply (5),**
- callForwardingUnconditional (6),**
- callForwardingOnBusy (7),**
- callForwardingOnNotReachable (8),**
- callWaiting (9),**
- callHold (10),**
- reverseCharging (11),**
- explicitCallTransfer (12),**
- callCompletionOnBusySubscriber (13),**
- adviceOfChargeOnStart (14),**
- adviceOfChargeAtEnd (15),**
- adviceOfChargeDuringCall (16),**
- timeDependentRouting (17),**
- callingPartingDependentRouting (18),**
- outgoingCallBarring (19),**
- incomingCallBarring (20)}**

**SuspendTimer ::= INTEGER (0..120)**

**TargetLineIdentifier ::= CHOICE {**

- individual [0] IMPLICIT OCTET STRING (SIZE (??..??)),**
- group [1] CHOICE {**

  - trunkGroupID [0] IMPLICIT INTEGER,**
  - privateFacilityID [1] IMPLICIT INTEGER,**
  - huntGroup [2] IMPLICIT OCTET STRING,**
  - routeIndex [3] IMPLICIT OCTET STRING}}**

**TerminalType ::= ENUMERATED {**

- unknown (0),**
- dialPulse (1),**
- dtmf (2),**
- isdn (3),**
- isdnNoDtmf (4),**
- spare (16)}**

**TimerID ::= ENUMERATED {**

- tssf (0)}**

**TimerValue ::= INTEGER (0..2147483647)**

**Tone ::= SEQUENCE {**

- toneID [0] IMPLICIT INTEGER (0..2147483647),**
- duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL}**

**ToneId ::= CHOICE {**

- local [0] IMPLICIT INTEGER (0..2147483647),**
- global [1] IMPLICIT OBJECT IDENTIFIER}**

**TraceInformation ::= SEQUENCE OF**

**SET {**

- scf [0] IMPLICIT OCTET STRING (SIZE (??..??))}**

```
TraceItem ::= SET {
    scf          [0] IMPLICIT OCTET STRING (SIZE (??..??))}
```

```
TravellingClassMark ::= OCTET STRING (SIZE (??..??))
```

```
TriggerDataIdentifier ::= SEQUENCE {
    triggerID [0] IMPLICIT ENUMERATED {
        origAttemptAuthorized (1),
        collectedInfo (2),
        analysedInformation (3),
        routeSelectFailure (4),
        oCalledPartyBusy (5),
        oNoAnswer (6),
        oAnswer (7),
        oMidCall (8),
        oDisconnect (9),
        oAbandon (10),
        termAttemptAuthorized (12),
        tBusy (13),
        tNoAnswer (14),
        tAnswer (15),
        tMidCall (16),
        tDisconnect (17),
        tAbandon (18),
        oTermSeized (19),
        oSuspended (20),
        tSuspended (21),
        origAttempt (22),
        termAttempt (23),
        oReAnswer (24),
        tReAnswer (25),
        facilitySelectedAndAvailable (26),
        callAccepted (27)},
    profileIdentifier [1] CHOICE {
        access [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        group [1] CHOICE {
            . trunkGroupID [0] IMPLICIT INTEGER,
            . privateFacilityID [1] IMPLICIT INTEGER,
            . huntGroup [2] IMPLICIT OCTET STRING,
            . routeIndex [3] IMPLICIT OCTET STRING},
        extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
            SEQUENCE {} OPTIONAL}
```

```
TriggerType ::= ENUMERATED {
    featureActivation (0),
    verticalServiceCode (1),
    customizedAccess (2),
    customizedIntercom (3),
    emergencyService (12),
    aFR . (13),
    sharedIOTrunk (14),
    offHookDelay (17),
    channelSetupPRI (18),
    tNoAnswer (25),
    tBusy (26),
    oCalledPartyBusy (27),
    oNoAnswer (29),
    originationAttemptAuthorized (30),
    oAnswer (31),
    oDisconnect (32),
```

```
termAttemptAuthorized (33),  
tAnswer (34),  
tDisconnect (35)}
```

```
UnavailableNetworkResource ::= ENUMERATED {  
    unavailableResources (0),  
    componentFailure (1),  
    basicCallProcessingException (2),  
    resourceStatusFailure (3),  
    endUserFailure (4)}
```

```
UserCredit ::= CHOICE {  
    currency SEQUENCE {  
        currency PrintableString (SIZE (3)),  
        amount INTEGER (0..??)},  
    units INTEGER (0..65536)}
```

```
UserInfo ::= SEQUENCE OF  
SEQUENCE {  
    infoToSend [0] CHOICE {  
        messageID [0] CHOICE {  
            elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),  
            text [1] IMPLICIT SEQUENCE {  
                messageContent [0] IMPLICIT IA5String (SIZE (??..??)),  
                attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},  
                elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF  
                INTEGER (0..2147483647),  
                variableMessage [30] IMPLICIT SEQUENCE {  
                    elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),  
                    variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF  
                        CHOICE {  
                            integer [0] IMPLICIT INTEGER (0..2147483647),  
                            number [1] IMPLICIT OCTET STRING (SIZE (??..??)),  
                            time [2] IMPLICIT OCTET STRING (SIZE (2)),  
                            date [3] IMPLICIT OCTET STRING (SIZE (3)),  
                            price [4] IMPLICIT OCTET STRING (SIZE (4))}},  
            toneId [1] CHOICE {  
                local [0] IMPLICIT INTEGER (0..2147483647),  
                global [1] IMPLICIT OBJECT IDENTIFIER},  
                displayInformation [2] IMPLICIT IA5String (SIZE (??..??)),  
            constraints [1] IMPLICIT SEQUENCE {  
                maximumNumberOfDigits [1] IMPLICIT INTEGER (1..127),  
                minimumNumberOfDigits [2] IMPLICIT INTEGER (1..127),  
                typeOfRequestedInfo [3] IMPLICIT ENUMERATED {  
                    numericString (0),  
                    characterString (1),  
                    iA5String (2)} DEFAULT numericString,  
                numberOfWorkRetries [4] IMPLICIT INTEGER (0..127) DEFAULT 0},  
            errorInfo [2] CHOICE {  
                messageID [0] CHOICE {  
                    elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),  
                    text [1] IMPLICIT SEQUENCE {  
                        messageContent [0] IMPLICIT IA5String (SIZE (??..??)),  
                        attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},  
                        elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF  
                        INTEGER (0..2147483647),  
                        variableMessage [30] IMPLICIT SEQUENCE {  
                            elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),  
                            variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF  
                                CHOICE {  
                                    integer [0] IMPLICIT INTEGER (0..2147483647),
```

```

.           .           .           number   [1] IMPLICIT OCTET STRING (SIZE (??..??)),
.           .           .           time     [2] IMPLICIT OCTET STRING (SIZE (2)),
.           .           .           date     [3] IMPLICIT OCTET STRING (SIZE (3)),
.           .           .           price    [4] IMPLICIT OCTET STRING (SIZE (4))}}}},
.           toneId [1] CHOICE {
.             local      [0] IMPLICIT INTEGER (0..2147483647),
.             global     [1] IMPLICIT OBJECT IDENTIFIER},
.           displayInformation [2] IMPLICIT IA5String (SIZE (??..??)))} OPTIONAL}

```

```

UserInformation ::= SEQUENCE {
  infoToSend [0] CHOICE {
    messageID [0] CHOICE {
      elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
      text       [1] IMPLICIT SEQUENCE {
        messageContent [0] IMPLICIT IA5String      (SIZE (??..??)),
        attributes    [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
      elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF
        INTEGER (0..2147483647),
      variableMessage [30] IMPLICIT SEQUENCE {
        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
        variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
          CHOICE {
            integer     [0] IMPLICIT INTEGER (0..2147483647),
            number      [1] IMPLICIT OCTET STRING (SIZE (??..??)),
            time       [2] IMPLICIT OCTET STRING (SIZE (2)),
            date       [3] IMPLICIT OCTET STRING (SIZE (3)),
            price      [4] IMPLICIT OCTET STRING (SIZE (4))}}}},
    toneId [1] CHOICE {
      local      [0] IMPLICIT INTEGER (0..2147483647),
      global     [1] IMPLICIT OBJECT IDENTIFIER},
    displayInformation [2] IMPLICIT IA5String (SIZE (??..??)))},
  constraints [1] IMPLICIT SEQUENCE {
    maximumNumberOfDigits [1] IMPLICIT INTEGER (1..127),
    minimumNumberOfDigits [2] IMPLICIT INTEGER (1..127),
    typeOfRequestedInfo [3] IMPLICIT ENUMERATED {
      numericString (0),
      characterString (1),
      iA5String     (2)} DEFAULT numericString ,
    numberOfAllowedRetries [4] IMPLICIT INTEGER (0..127) DEFAULT 0},
  errorInfo [2] CHOICE {
    messageID [0] CHOICE {
      elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
      text       [1] IMPLICIT SEQUENCE {
        messageContent [0] IMPLICIT IA5String      (SIZE (??..??)),
        attributes    [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
      elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF
        INTEGER (0..2147483647),
      variableMessage [30] IMPLICIT SEQUENCE {
        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
        variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
          CHOICE {
            integer     [0] IMPLICIT INTEGER (0..2147483647),
            number      [1] IMPLICIT OCTET STRING (SIZE (??..??)),
            time       [2] IMPLICIT OCTET STRING (SIZE (2)),
            date       [3] IMPLICIT OCTET STRING (SIZE (3)),
            price      [4] IMPLICIT OCTET STRING (SIZE (4))}}}},
    toneId [1] CHOICE {
      local      [0] IMPLICIT INTEGER (0..2147483647),
      global     [1] IMPLICIT OBJECT IDENTIFIER},
    displayInformation [2] IMPLICIT IA5String (SIZE (??..??)))} OPTIONAL}

```

```

UserInteractionModes ::= BIT STRING {
    voiceMessage (0),
    tone (1),
    display (2)}

USIInformation ::= OCTET STRING (SIZE (??..??))

USIMonitorMode ::= ENUMERATED {
    monitoringActive (0),
    monitoringInactive (1)}

USIServiceIndicator ::= OCTET STRING (SIZE (??..??))

VariablePart ::= CHOICE {
    integer [0] IMPLICIT INTEGER (0..2147483647),
    number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
    time [2] IMPLICIT OCTET STRING (SIZE (2)),
    date [3] IMPLICIT OCTET STRING (SIZE (3)),
    price [4] IMPLICIT OCTET STRING (SIZE (4))}

highLayerCompatibilityLength INTEGER ::= 2
minAChBillingChargingLength INTEGER ::= ??
maxAChBillingChargingLength INTEGER ::= ??
minAttributesLength INTEGER ::= ??
maxAttributesLength INTEGER ::= ??
minBackwardGVNSLength INTEGER ::= ??
maxBackwardGVNSLength INTEGER ::= ??
maxBearerCapabilityLength INTEGER ::= ??
minCalledPartyNumberLength INTEGER ::= ??
maxCalledPartyNumberLength INTEGER ::= ??
minCallingPartyNumberLength INTEGER ::= ??
maxCallingPartyNumberLength INTEGER ::= ??
minCallResultLength INTEGER ::= ??
maxCallResultLength INTEGER ::= ??
minCauseLength INTEGER ::= 2
maxCauseLength INTEGER ::= ??
minDigitsLength INTEGER ::= ??
maxDigitsLength INTEGER ::= ??
minDisplayInformationLength INTEGER ::= ??
maxDisplayInformationLength INTEGER ::= ??
minEventSpecificInformationChargingLength INTEGER ::= ??

```

**maxEventSpecificInformationChargingLength** INTEGER ::= ??  
**minEventTypeChargingLength** INTEGER ::= ??  
**maxEventTypeChargingLength** INTEGER ::= ??  
**minFCIBillingChargingLength** INTEGER ::= ??  
**maxFCIBillingChargingLength** INTEGER ::= ??  
**minForwardGVNSLength** INTEGER ::= ??  
**maxForwardGVNSLength** INTEGER ::= ??  
**minGenericNameLength** INTEGER ::= ??  
**maxGenericNameLength** INTEGER ::= ??  
**minGenericNumberLength** INTEGER ::= ??  
**maxGenericNumberLength** INTEGER ::= ??  
**maxInitialTimeInterval** INTEGER ::= ??  
**maxINServiceCompatibilityIndLength** INTEGER ::= ??  
**minIPAvailableLength** INTEGER ::= ??  
**maxIPAvailableLength** INTEGER ::= ??  
**minIPSSPCapabilitiesLength** INTEGER ::= ??  
**maxIPSSPCapabilitiesLength** INTEGER ::= ??  
**minLocationNumberLength** INTEGER ::= ??  
**maxLocationNumberLength** INTEGER ::= ??  
**minMailBoxIDLength** INTEGER ::= ??  
**maxMailBoxIDLength** INTEGER ::= ??  
**minMessageContentLength** INTEGER ::= ??  
**maxMessageContentLength** INTEGER ::= ??  
**minMidCallControlInfoNum** INTEGER ::= ??  
**maxMidCallControlInfoNum** INTEGER ::= ??  
**minOriginalCalledPartyIDLength** INTEGER ::= ??  
**maxOriginalCalledPartyIDLength** INTEGER ::= ??  
**minReasonLength** INTEGER ::= ??  
**maxReasonLength** INTEGER ::= ??  
**maxRecordedMessageUnits** INTEGER ::= ??

**maxRecordingTime** INTEGER ::= ??  
**minRedirectingPartyIDLength** INTEGER ::= ??  
**maxRedirectingPartyIDLength** INTEGER ::= ??  
**minRouteListLength** INTEGER ::= ??  
**maxRouteListLength** INTEGER ::= ??  
**minScfIDLength** INTEGER ::= ??  
**maxScfIDLength** INTEGER ::= ??  
**minSCIBillingChargingLength** INTEGER ::= ??  
**maxSCIBillingChargingLength** INTEGER ::= ??  
**minServiceInteractionIndicatorsLength** INTEGER ::= ??  
**maxServiceInteractionIndicatorsLength** INTEGER ::= ??  
**minSFBillingChargingLength** INTEGER ::= ??  
**maxSFBillingChargingLength** INTEGER ::= ??  
**minUSIInformationLength** INTEGER ::= ??  
**maxUSIInformationLength** INTEGER ::= ??  
**minUSIServiceIndicatorLength** INTEGER ::= ??  
**maxUSIServiceIndicatorLength** INTEGER ::= ??  
**numOfBCSMEvents** INTEGER ::= ??  
**numOfBCUSMEvents** INTEGER ::= ??  
**numOfChargingEvents** INTEGER ::= ??  
**numOfCounters** INTEGER ::= 100  
**numOfCSAs** INTEGER ::= ??  
**numOfCSs** INTEGER ::= ??  
**numOfExtensions** INTEGER ::= ??  
**numOfInfoItems** INTEGER ::= 5  
**numOfGenericNumbers** INTEGER ::= ??  
**numOfLegs** INTEGER ::= ??  
**numOfMessageIDs** INTEGER ::= ??  
**numOfRecordedMessageIDs** INTEGER ::= ??  
**maxCreditUnit** INTEGER ::= 65536  
**maxInitialUnitIncrement** INTEGER ::= ??

```
maxScalingFactor INTEGER ::= ??  
maxTimePerInterval INTEGER ::= ??  
maxUnitsPerDataInterval INTEGER ::= ??  
maxUnitsPerInterval INTEGER ::= ??  
maxSegmentsPerDataInterval INTEGER ::= ??  
ub-maxUserCredit INTEGER ::= ??  
maxAmount INTEGER ::= ??  
numOfInServiceCompatibilityIndLength INTEGER ::= ??  
minReceivedInformationLength INTEGER ::= ??  
maxReceivedInformationLength INTEGER ::= ??  
minRequestedUTSINum INTEGER ::= ??  
maxRequestedUTSINum INTEGER ::= ??  
minScfAddressLength INTEGER ::= ??  
maxScfAddressLength INTEGER ::= ??  
ub-nbCall INTEGER ::= ??  
END
```

## APÉNDICE II

### Modelado de datos

#### II.1 Introducción

##### II.1.1 Objetivo y ámbito

En este apéndice se describe un modelado de objetos que puede utilizarse para la transferencia de perfiles del servicio interredes (ISPT, *inter-network service profile transfer*).

##### II.1.2 Hipótesis

###### II.1.2.1 Servicio de movilidad e identificadores de abonado

###### II.1.2.1.1 Identificador único de abonado al servicio de movilidad

Un abonado al servicio de movilidad debe tener un identificador numérico único para distinguirse de otros abonados del mismo proveedor de servicio de movilidad. Los identificadores de abonado son localmente significativos. Es decir, los proveedores de servicio no necesitan conocer los identificadores únicos de abonados de otros proveedores de servicio.

Sería conveniente establecer mediante acuerdo una longitud máxima, aunque dentro de ese límite la longitud de un identificador de abonado es una cuestión local.

### **II.1.2.1.2 Prefijo de proveedor de servicio de movilidad**

Cada proveedor de servicio de movilidad debe tener un identificador numérico único (prefijo). Los prefijos de los proveedores de servicio son globalmente significativos. Es decir, los proveedores de servicio de movilidad cooperantes deben conocer los identificadores numéricos de los demás.

Sería conveniente establecer mediante acuerdo una longitud máxima.

Al utilizar un identificador de proveedor de servicio como prefijo de cada identificador único de abonado local todos los abonados al servicio de movilidad podrían ser identificados individualmente y sin equívocos.

### **II.1.2.1.3 Subprefijo de proveedor de servicio de movilidad**

Si un proveedor de servicio de movilidad necesita estar en condiciones de agrupar lógicamente a sus abonados, por ejemplo si se utilizan múltiples DSA, puede emplearse un subprefijo del proveedor de servicio. Los subprefijos de proveedor de servicio son localmente significativos. Es decir, los proveedores de servicio de movilidad cooperantes necesitan conocer los identificadores de subprefijo de los demás.

Sería conveniente establecer mediante acuerdo una longitud máxima, aunque dentro de ese límite la longitud de un subprefijo de proveedor de servicio, es una cuestión local.

En este apéndice no se define la utilización de subprefijos adicionales. Se considera innecesario, aunque no sería difícil definirlos y aplicarlos.

### **II.1.2.1.4 Codificación**

El medio por el cual se revela a una red visitada un identificador único de usuario itinerante, un prefijo y un subprefijo de proveedor de servicio, queda fuera del alcance del presente apéndice. A los efectos de este apéndice, se supone algún tipo de codificación, por ejemplo del identificador de objeto, su longitud y su valor, de modo que puedan reconocerse individualmente un identificador único de usuario y así el prefijo y el subprefijo del proveedor de servicio.

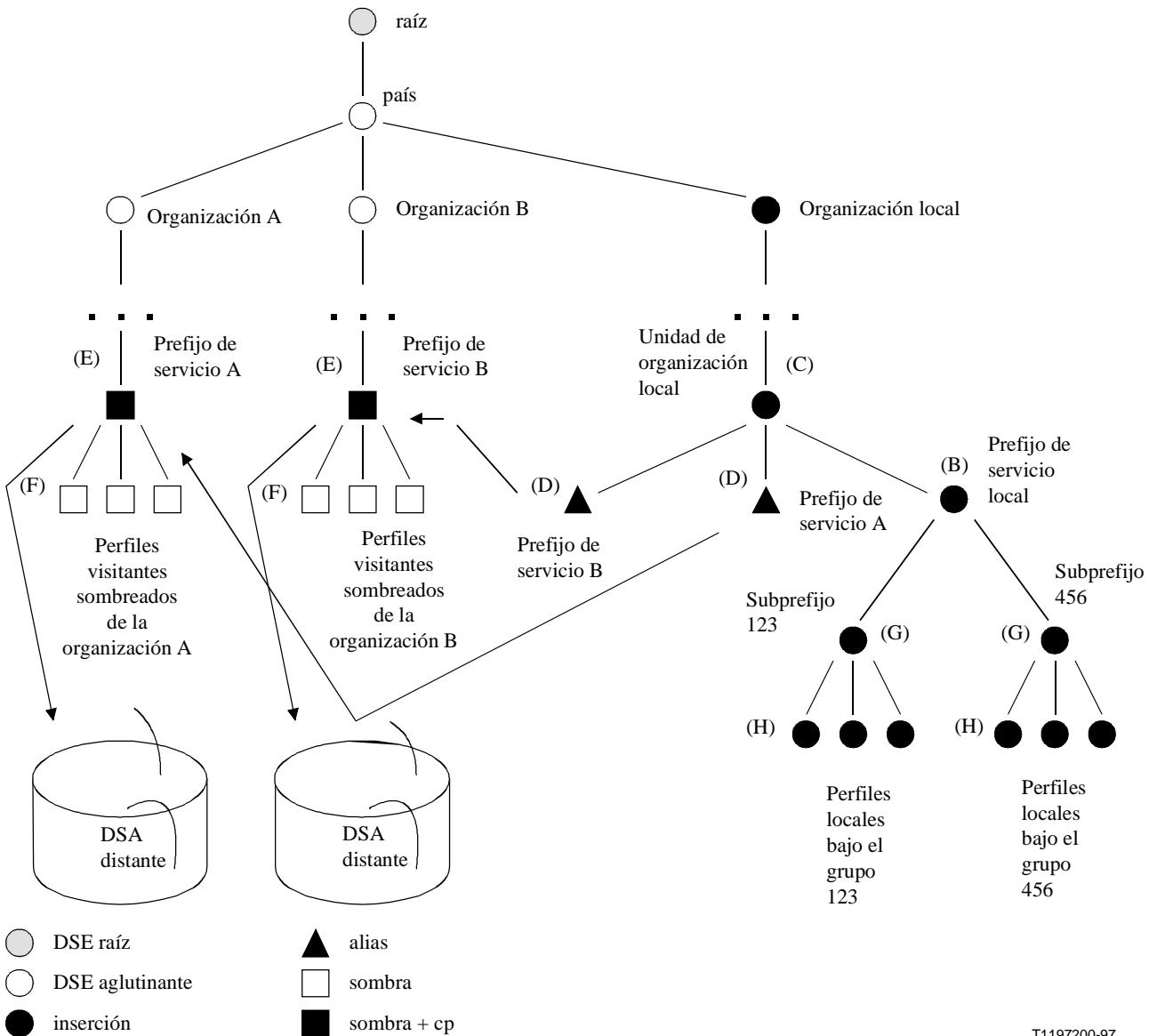
## **II.1.2.2 Revelación de inserciones de perfil**

No es conveniente que la lista completa de inserciones de abonado de un proveedor de servicio esté a la disposición de los demás proveedores de servicio. Deberían revelarse solamente los perfiles de los usuarios itinerantes.

## **II.2 Esquemas del árbol de información de directorio (DIT, *directory information tree*)**

### **II.2.1 Árbol de información del directorio de la Recomendación X.500**

El DIT (*directory information tree*) de la Recomendación X.500 que se muestra en la figura II-1 ilustra el esquema DIT descrito en este apéndice.



**Figura II.1/Q.1228 – Ejemplo de DIT local del proveedor de servicio de movilidad**

### II.2.1.1 Ubicación de los perfiles locales

Se utilizará un prefijo de proveedor de servicio para denominar el nodo [véase el nodo etiquetado (B) en la figura II.1] en el DIT local del proveedor de servicio bajo el que están almacenados todos los perfiles de abonados locales. La inserción del prefijo del proveedor de servicio contendrá una clase de objeto cuyo valor es `inMobilityServicePrefix` (prefijo del servicio de movilidad en la red inteligente).

Los perfiles de usuario pueden ser agrupados por subprefijos de proveedor de servicio. Si se utiliza un subprefijo, se empleará como un nombre distinguido relativo adicional, y nodo [nodos etiquetados (G)] en el DIT, entre un identificador único de usuario y su prefijo de proveedor de servicio. Las inserciones de subprefijo de proveedor de servicio contendrán, si así procede, una clase de objeto cuyo valor es `inMobilityServiceSubPrefix` (subprefijo del servicio de movilidad en la red inteligente). Las inserciones de subprefijo se denominarán utilizando los identificadores de subprefijo de proveedor del servicio.

Las inserciones de perfil local [nodos etiquetados (H)] contendrán una clase de objeto cuyo valor es inMobilityUserProfile (perfil de usuario de movilidad en la red inteligente). Las inserciones de abonado local se denominarán utilizando su identificador de servicio de movilidad único.

#### **II.2.1.2 Ubicación de los perfiles de abonados distantes**

La ubicación de un perfil de abonado del proveedor del servicio debe estar a disposición de los demás proveedores de servicio cooperantes. Las inserciones [nodos etiquetados (E)] superiores a perfiles almacenados en los DSA distantes, deben estar sombreadas en el DSA local. Mientras esta inserción de DSE sombra tenga un conocimiento subordinado incompleto, la red visitada puede generar una referencia de continuación desde el atributo supplierKnowledge (conocimiento del suministrador) hallado en dicha inserción. La referencia de continuación apuntará al DSA de base del usuario itinerante en donde están ubicados los perfiles reales.

#### **II.2.1.3 Ubicación de los perfiles de abonados visitantes locales**

Los abonados visitantes locales tendrán sus inserciones de perfiles [nodo etiquetado (F)] sombreadas bajo el nodo anteriormente descrito [nodos etiquetados (E)], que corresponde a su proveedor de servicio de base.

#### **II.2.1.4 Disminución del flujo de mensajes**

Cuando un usuario se desplaza a otra red, la red visitada debe modificar el perfil del usuario itinerante en la red de base del usuario. Esto hará que el perfil de usuario se transfiera a la red visitada.

Para modificar el perfil de usuario, la red visitada debe determinar el DN del usuario itinerante. Para disminuir el flujo de mensajes, es posible que este nombre distinguido pueda resolverse sin utilizar las operaciones de búsqueda del DAP. El establecimiento de la correspondencia entre identificadores de abonado y números de directorio y los DN mediante la utilización de planes de numeración de longitud fija.

En cuanto a las hipótesis mencionadas anteriormente, se revelan a la red visitada un identificador y un prefijo (y subprefijo, si así procede) únicos de usuario itinerante. Identificador y prefijo (y subprefijo) que pueden utilizarse como RDN componentes en el DN del usuario itinerante. Es posible generar un DN preliminar agregando estos RDN al DN del nodo superior a la inserción de prefijo de la red de base [nodo (B)].

Se consideran inserciones alias [nodos etiquetados (D)] para cada proveedor de servicio cooperante, adyacente al nodo bajo el cual se almacenan los perfiles de abonado del servicio local [nodo (B)]. Se supone además que estas inserciones alias reciben su denominación utilizando números de prefijo del proveedor de servicio y se utilizan para señalar su nodo correspondiente (nodos etiquetados (E)) ubicado en otra parte del DIT local.

Cuando se utilice en una operación DAP, el DN preliminar del usuario itinerante será desreferenciado para señalar el nodo en el DIT que es una sombra del prefijo de la red de base del usuario itinerante. Mientras esta inserción DSE sombra tenga un conocimiento subordinado incompleto, la red visitada puede generar una referencia de continuación desde el atributo supplierKnowledge hallado en dicha inserción. La referencia de continuación apuntará al DSA de base del usuario itinerante en donde están ubicados los perfiles reales. Esta operación se encadenará a continuación con el DSA de base del usuario itinerante.

## II.2.2 Clases de objetos

### II.2.2.1 Perfil de usuario de movilidad en la red inteligente

Se ha definido la clase de objeto inMobilityUserProfile (perfil de usuario de movilidad en la red inteligente) para almacenar información de perfiles. La siguiente definición ASN.1 puede utilizarse como punto de partida para describir la clase de objeto inMobilityUserProfile:

```
inMobilityUserProfile   OBJECT-CLASS ::= {  
    SUBCLASS OF { top }  
    MUST CONTAIN { inMobilityID |  
                  inMobilityPIN |  
                  <other mandatory attributes> }  
    MAY CONTAIN { <optional attributes> }  
    ID           Id-oc-inMobilityUserProfile }
```

El atributo inMobilityID es el atributo distinguido.

Las inserciones de este tipo se ubicarán únicamente bajo una inserción de tipo inMobilityServiceProvider o inMobilitySubscriberGroup.

### II.2.2.2 Proveedor de servicio de movilidad en la red inteligente

Se ha creado esta clase de objeto para definir un nodo bajo el cual pueden almacenarse los objetos de tipo inMobilityUserProfile (perfil de usuario de movilidad en la red inteligente) e inMobilitySubscriberGroup (grupo de abonados de movilidad en la red inteligente). La siguiente definición ASN.1 describe la clase de objeto inMobilityServiceProvider:

```
inMobilityServiceProvider   OBJECT-CLASS ::= {  
    SUBCLASS OF { top }  
    MUST CONTAIN { inMobilityPrefix | <other mandatory attributes> }  
    MAY CONTAIN { <optional attributes> }  
    ID           Id-oc-inMobilityServiceProvider }
```

El atributo inMobilityPrefix es el atributo distinguido.

Las inserciones de este tipo pueden contener una referencia subordinada no específica (NSSR) que apunta a las inserciones de perfil, o inserciones de subprefijos, si así procede, almacenadas en un DSA distante.

Las inserciones subordinadas a esta clase de objeto deben ser del tipo inMobilityUserProfile o inMobilitySubscriberGroup.

### II.2.2.3 Grupo de abonados de movilidad en la red inteligente

La clase de objeto inMobilitySubscriberGroup (grupo de abonados de movilidad de la red inteligente) se ha creado para definir un nodo bajo el cual pueden almacenarse los objetos de tipo inMobilityUserProfile. La siguiente definición ASN.1 describe la clase de objeto inMobilityServiceProvider:

```
inMobilitySubscriberGroup   OBJECT-CLASS ::= {  
    SUBCLASS OF { top }  
    MUST CONTAIN { inMobilitySubPrefix | <other mandatory attributes> }  
    MAY CONTAIN { <optional attributes> }  
    ID           Id-oc-inMobilitySubscriberGroup }
```

El atributo inMobilitySubPrefix es el atributo distinguido.

Las inserciones subordinadas a esta clase de objeto deben ser del tipo inMobilityUserProfile.

## II.2.3 Tipos de atributos

### II.2.3.1 Identificador de movilidad en la red

El atributo inMobilityID (identificador de movilidad en la red) se utiliza para distinguir inequívocamente entre los usuarios del servicio de movilidad de un proveedor o un grupo de abonados de un determinado servicio.

La definición ASN.1 para inMobilityID es la siguiente:

```
inMobilityID          ATTRIBUTE ::= {  
    WITH SYNTAX      Digits (SIZE(lb-inMobilityID..ub-inMobilityID))  
    EQUALITY MATCHING RULE  
    ID               octetStringMatch  
                      id-at-inMobilityID}
```

### II.2.3.2 PIN de movilidad en la red inteligente

El atributo inMobilityPIN (PIN de movilidad en la red inteligente) se utiliza para almacenar el PIN de un usuario del servicio de movilidad.

```
inMobilityPIN         ATTRIBUTE ::= {  
    WITH SYNTAX      userPassword (SIZE lbinMobilityPIN..ubinMobilityPIN)  
    ID               id-at-inMobilityPIN}
```

### II.2.3.3 Prefijo de movilidad en la red inteligente

El atributo inMobilityPrefix (prefijo de movilidad en la red inteligente) distingue inequívocamente entre proveedores del servicio de movilidad.

```
inMobilityPrefix       ATTRIBUTE ::= {  
    WITH SYNTAX      Digits (SIZE(lb-inMobilityPrefix..ub-inMobilityPrefix))  
    EQUALITY MATCHING RULE  
    ID               octetStringMatch  
                      id-at-inMobilityPrefix}
```

### II.2.3.4 Subprefijo de movilidad en la red inteligente

El atributo inMobilitySubPrefix (subprefijo de movilidad en la red inteligente) se utiliza para agrupar los abonados del servicio de movilidad.

```
inMobilitySubPrefix    ATTRIBUTE ::= {  
    WITH SYNTAX      Digits (SIZE(lb-inMobilitySubPrefix..  
                                ub-inMobilitySubPrefix))  
    EQUALITY MATCHING RULE  
    ID               octetStringMatch  
                      id-at-inMobilitySubPrefix}
```

## II.2.4 Definición de la estructura del árbol de información de directorio (DIT)

### II.2.4.1 Formas de nombres

Una forma de nombre especifica el atributo que debe utilizarse como RDN para una clase de objeto específico.

#### II.2.4.1.1 inMobilityUserProfileNameForm (forma de nombre del perfil de usuario de movilidad en la red inteligente)

La siguiente definición de forma de nombre indica que inMobilityID es el atributo distinguido permitido para la clase de objeto inMobilityUserProfile.

```
inMobilityUserProfileNameForm NAME-FORM ::= {  
    WITH ATTRIBUTES  
    ID               inMobilityID  
                      id-nf-inMobilityUserProfileNameForm}
```

## II.2.4.1.2 inMobilityServiceProviderNameForm (forma de nombre de proveedor de servicio de movilidad en la red inteligente)

La siguiente definición de forma de nombre indica que inMobilityPrefix es el atributo distinguido permitido para la clase de objeto inMobilityServiceProvider.

<b>inMobilityServiceProviderNameForm</b> NAMES WITH ATTRIBUTES ID	NAME-FORM ::= { inMobilityServiceProvider inMobilityPrefix id-nf-inMobilityServiceProviderNameForm}
--	--

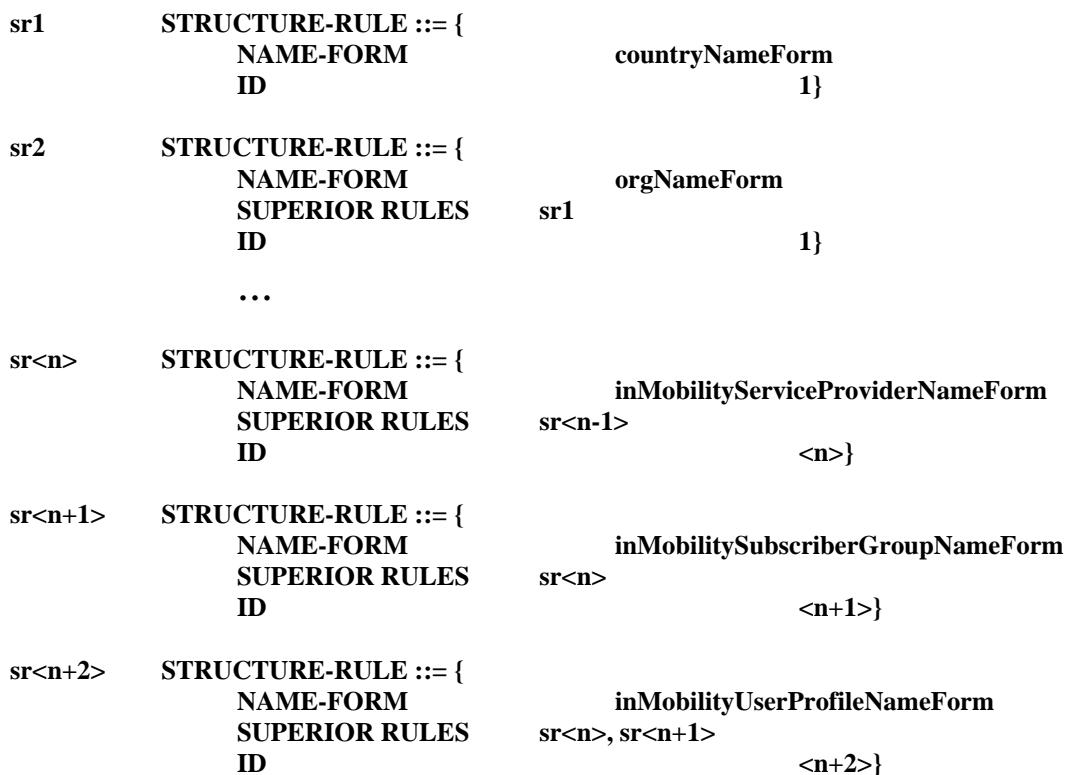
## II.2.4.1.3 inMobilitySubscriberGroup (grupo de abonados de movilidad en la red inteligente)

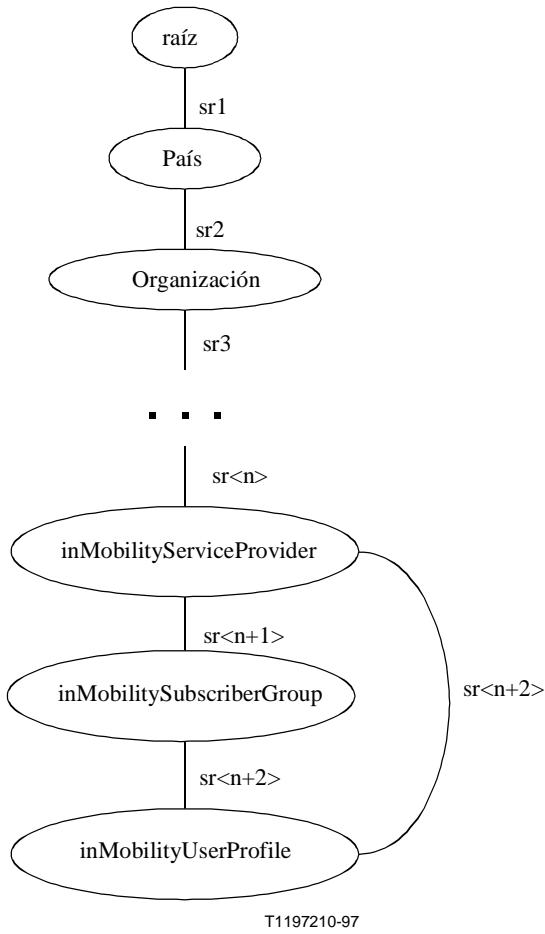
La siguiente definición de forma de nombre indica que inMobilitySubPrefix es el atributo distinguido permitido para la clase de objeto inMobilitySubscriberGroup.

<b>inMobilitySubscriberGroupNameForm</b> NAMES WITH ATTRIBUTES ID	NAME-FORM ::= { inMobilitySubscriberGroup inMobilitySubPrefix id-nf-inMobilitySubscriberGroupNameForm}
--	---

## II.2.4.2 Reglas de estructura

Las reglas de estructura especifican las inserciones subordinadas y superiores permitidas en un DIT. Las siguientes reglas de estructura ilustradas en la figura II.2, pueden utilizarse como base para definir las reglas de estructura requeridas para el servicio de movilidad:





**Figura II.2/Q.1228 – Reglas de estructura**

#### II.2.4.3 Asignación de identificadores de objeto

Las siguientes asignaciones de identificadores de objeto puede utilizarse como punto de partida para identificar objetos del servicio de movilidad en la serie de Recomendaciones X.500.

<b>id-at-inMobilityID</b>	<b>OBJECT IDENTIFIER ::= {id-at-inMobility 0}</b>
<b>id-at-inMobilityPIN</b>	<b>OBJECT IDENTIFIER ::= {id-at-inMobility 1}</b>
<b>id-at-inMobilityPrefix</b>	<b>OBJECT IDENTIFIER ::= {id-at-inMobility 2}</b>
<b>id-at-inMobilitySubPrefix</b>	<b>OBJECT IDENTIFIER ::= {id-at-inMobility 3}</b>
<b>id-oc-inMobilityUserProfile</b>	<b>OBJECT IDENTIFIER ::= {id-oc-inMobility 0}</b>
<b>id-oc-inMobilityServiceProvider</b>	<b>OBJECT IDENTIFIER ::= {id-oc-inMobility 1}</b>
<b>id-oc-inMobilitySubscriberGroup</b>	<b>OBJECT IDENTIFIER ::= {id-oc-inMobility 2}</b>
<b>id-nf-inMobilityUserProfileNameForm</b>	<b>OBJECT IDENTIFIER ::= {id-nf-inMobility 0}</b>
<b>id-nf-inMobilityServiceProviderNameForm</b>	<b>OBJECT IDENTIFIER ::= {id-nf-inMobility 1}</b>
<b>id-nf-inMobilitySubscriberGroupNameForm</b>	<b>OBJECT IDENTIFIER ::= {id-nf-inMobility 2}</b>

## APÉNDICE III

### Ejemplos de algoritmos SPKM para el CS-2 de la red inteligente

#### III.1 Consideraciones generales

En el SPKM se utilizan un cierto número de tipos de algoritmo. El presente apéndice describe cada uno de ellos, junto con su finalidad y un conjunto de ejemplos específicos. Para asegurar por lo menos un nivel mínimo de interfuncionamiento entre diversas realizaciones del SPKM para el CS-2 de la red inteligente, se especifica uno de los algoritmos de integridad como OBLIGATORIO; los ejemplos restantes (y cualquier otro algoritmo), pueden ser admitidos opcionalmente por una realización SPKM dada y, por ello, este apéndice especifica algunos algoritmos como RECOMENDADOS.

El SPKM utiliza los términos "iniciador" y "objetivo". En el caso del CS-2 de la red inteligente, puede hacerse referencia al término "objetivo" como "consumidor".

#### III.2 Algoritmo de integridad (I-ALG)

La finalidad de este algoritmo es asegurar que un mensaje no ha sido alterado de ninguna manera una vez construido por el emisor legitimado. Dependiendo del algoritmo utilizado, la aplicación de este algoritmo puede proporcionar también autenticidad y soporte de no rechazo para el mensaje.

##### III.2.1 Ejemplo–1

```
md5WithRSAEncryption OBJECT IDENTIFIER ::= {  
    iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1)pkcs-1(1) 4  
}
```

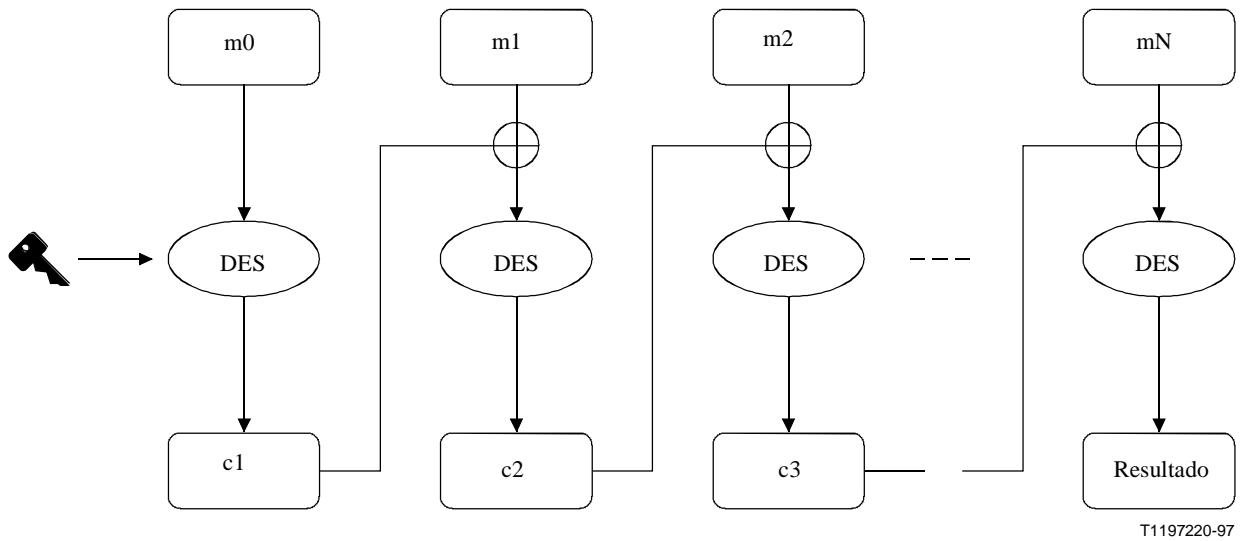
Este algoritmo (OBLIGATORIO) proporciona integridad y autenticidad de datos y soporte de no rechazo mediante el cálculo de una firma RSA en el troceado MD5 de esos datos.

Se hace notar que, puesto que este es el único algoritmo de integridad/autenticidad especificado como obligatorio, en la actualidad se estipula también por razones de interfuncionamiento que md5WithRSA sea el algoritmo utilizado para firmar todos los testigos de establecimiento de contexto que se firman en lugar del MACed. En versiones futuras de este apéndice, se pueden especificar algoritmos alternativos o adicionales obligatorios, de manera que pueda eliminarse esa estipulación sobre los testigos de establecimiento de contexto.

##### III.2.2 Ejemplo–2

```
DES-MAC OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 10  
}  
    -- carries length in bits of the MAC as an INTEGER parameter, constrained to  
    -- multiples of eight from 16 to 64
```

Este algoritmo (RECOMENDADO) proporciona integridad mediante el cálculo de un DES-MAC en esos datos como se ilustra en la figura III.1.



Texto claro  $M = m_1 \text{II} m_2 \text{II} m_3 \text{II} \dots \text{II} m_N$ ,  
 II: concatenación  
 $m_0$ : Valor inicial (por ejemplo, todos "0")

**Figura III.1/Q.1228 – Mecanismo DES-MAC**

### III.2.3 Ejemplo–3

```
md5-DES-CBC OBJECT IDENTIFIER ::= {
  iso(1) identified-organization(3) dod(6) internet(1)
  security(5) integrity(3) md5-DES-CBC(1)
}
```

Este algoritmo proporciona integridad de datos mediante la encriptación, utilizando el DES-CBC, es decir el troceado MD5 "embrollado" (*confounded*) de los datos. En la práctica, este proceso será normalmente más rápido que el de calcular un DES-MAC a menos que los datos de entrada sean extremadamente cortos (por ejemplo, de unos pocos bytes). Cabe señalar que sin el embrollador (*confounder*) la fortaleza de este mecanismo de integridad es (a lo sumo) igual a la del DES frente a un ataque de texto claro conocido.

### III.2.4 Ejemplo–4

```
sum64-DES-CBC OBJECT IDENTIFIER ::= {
  iso(1) identified-organization(3) dod(6) internet(1)
  security(5) integrity(3) sum64-DES-CBC(2)
}
```

Este algoritmo proporciona integridad de datos mediante la encriptación, utilizando el DES-CBC, la concatenación de los datos embrollados y la suma de todos los bloques de datos de entrada (la suma calculada utilizando el módulo de adición  $2^{**64} - 1$ ). Así pues, en este algoritmo, la encriptación es un requisito para asegurar la integridad.

## III.3 Algoritmo de confidencialidad (C-ALG)

Este algoritmo simétrico se utiliza para generar los datos encriptados.

### **III.3.1 Ejemplo–1**

```
DES-CBC OBJECT IDENTIFIER ::= {
  iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 7
  }      -- carries IV (OCTET STRING) as a parameter; this (optional) parameter is
         -- unused in SPKM due to the use of confounding
```

Este algoritmo es RECOMENDADO.

### **III.4 Algoritmo de establecimiento de clave (K-ALG)**

Este algoritmo se utiliza para establecer una clave simétrica que será utilizada por el iniciador y el objetivo en el contexto (de seguridad) establecido. Las claves utilizadas para C-ALG y cualquiera I-ALG por claves (por ejemplo, DES-MAC) se derivan de esta clave de contexto. El establecimiento de las claves se efectúa en el marco del intercambio de autenticación de la Recomendación X.509, por lo que la clave simétrica compartida resultante es una clave autenticada.

#### **III.4.1 Ejemplo–1**

```
RSAEncryption OBJECT IDENTIFIER ::= {
  iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) 1
  }
```

En este algoritmo (OBLIGATORIO), la clave de contexto es generada por el iniciador, que la encripta con la clave pública RSA del objetivo, y la envía al objetivo. No es preciso que el objetivo responda al iniciador para que la clave sea establecida.

#### **III.4.2 Ejemplo–2**

```
id-rsa-key-transport OBJECT IDENTIFIER ::= {
  iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 22
  }
```

Similar a la encriptación RSA, la información autenticante del origen es también encriptada con la clave pública RSA del objetivo.

#### **III.4.3 Ejemplo–3**

```
dhKeyAgreement OBJECT IDENTIFIER ::= {
  iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-3(3) 1
  }
```

En este algoritmo, la clave de contexto es generada conjuntamente por el iniciador y el objetivo utilizando el algoritmo de establecimiento de clave Diffie-Hellman. El objetivo debe responder por tanto al iniciador para que la clave sea establecida.

### **III.5 Función unidireccional (O-ALG) para el algoritmo de derivación de subclaves**

Una vez establecido una clave de contexto mediante el K-ALG negociado, el iniciador y el objetivo han de poder derivar un conjunto de subclaves para los diversos C-ALG e I-ALG por claves admitidos en el contexto. La lista (ordenada) de los algoritmos C-ALG convenidos debe numerarse de forma consecutiva de tal modo que el primero (algoritmo "por defecto") tenga el número "0", el siguiente el "1", y así sucesivamente. La numeración de la lista (ordenada) de los algoritmos I-ALG convenidos debe ser idéntica. La clave de contexto, por último, debe ser una cadena binaria de longitud arbitraria "M", sujeta a la siguiente limitación:  $L \leq M \leq U$  (donde el límite inferior "L" es la longitud en bits de la clave más larga que necesita cualquier algoritmo C-ALG convenido o I-ALG por claves convenido, y el límite superior "U" es el mayor tamaño en bits que se adaptará a los parámetros del K-ALG).

Por ejemplo, si DES y DES triple de dos claves son los algoritmos de confidencialidad negociados, y DES-MAC es el algoritmo de integridad por claves negociado (se señala que las firmas digitales no utilizan una clave de contexto), la clave de contexto debe tener una longitud mínima de 112 bits. Si el algoritmo K-ALG que se utiliza es el RSAEncryption de 512 bits, el originador puede generar aleatoriamente una clave de contexto de cualquier longitud mayor, hasta 424 bits. El objetivo puede determinar la longitud que ha sido escogida eliminando los bytes de relleno durante la operación de decrptación de RSA. Por otra parte, si dhKeyAgreement es el algoritmo K-ALG en uso la clave de contexto es el resultado del cálculo de Diffie-Hellman (con excepción del byte de orden superior, que se descarta por razones de seguridad), de modo tal que su longitud es la del módulo de Diffie-Hellman, p, menos 8 bits.

El algoritmo de derivación de una subclave de k bits se especifica como sigue:

rightmost\_k\_bits (OWF(context\_key || x || n || s || context\_key))

donde,

- "x" es el carácter "C" ASCII (0x43) si la subclave es para un algoritmo de confidencialidad o el carácter "I" ASCII (0x49) si la subclave es para un algoritmo de integridad por claves;
- "n" es el número del algoritmo en la lista convenida apropiada para el contexto [el carácter "0" ASCII (0x30), "1" (0x31), y así sucesivamente];
- "s" es la "etapa" de procesamiento – siempre el carácter "0" ASCII (0x30), salvo que "k" sea mayor que el tamaño de salida de la función unidireccional (OWF, *one-way function*), en cuyo caso la OWF se calcula repetidamente con valores ASCII crecientes de la "etapa" (estando cada salida OWF concatenada al extremo de las salidas OWF previas), hasta que se hayan generado "k" bits;
- "||" es la operación de concatenación; y
- "OWF" es toda función unidireccional apropiada.

### III.5.1 Ejemplo-1

```
MD5 OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) US(840) rsadsi(113549) digestAlgorithm(2) 5
}
```

Este algoritmo es OBLIGATORIO.

```
SHA OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 18
}
```

Se reconoce que las funciones de troceado existentes pueden no satisfacer todas las propiedades requeridas de las OWF. Esta es la razón por la que se permite la negociación de la OWF O-ALG durante el proceso de establecimiento de contexto, ya que de este modo se puede dar cabida fácilmente a las futuras mejoras en el diseño de la OWF. Por ejemplo, en algunos entornos una técnica OWF preferida podría ser un algoritmo de encriptación que encripta la entrada especificada anteriormente utilizando la clave de contexto (context\_key) como la clave de encriptación.



## **SERIES DE RECOMENDACIONES DEL UIT-T**

- Serie A Organización del trabajo del UIT-T
- Serie B Medios de expresión: definiciones, símbolos, clasificación
- Serie C Estadísticas generales de telecomunicaciones
- Serie D Principios generales de tarificación
- Serie E Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
- Serie F Servicios de telecomunicación no telefónicos
- Serie G Sistemas y medios de transmisión, sistemas y redes digitales
- Serie H Sistemas audiovisuales y multimedios
- Serie I Red digital de servicios integrados
- Serie J Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
- Serie K Protección contra las interferencias
- Serie L Construcción, instalación y protección de los cables y otros elementos de planta exterior
- Serie M RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
- Serie N Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
- Serie O Especificaciones de los aparatos de medida
- Serie P Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
- Serie Q Conmutación y señalización**
- Serie R Transmisión telegráfica
- Serie S Equipos terminales para servicios de telegrafía
- Serie T Terminales para servicios de telemática
- Serie U Conmutación telegráfica
- Serie V Comunicación de datos por la red telefónica
- Serie X Redes de datos y comunicación entre sistemas abiertos
- Serie Y Infraestructura mundial de la información
- Serie Z Lenguajes de programación