



INTERNATIONAL TELECOMMUNICATION UNION

# ITU-T

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

# Q.1228

## Fascicle 4/5

(09/97)

SERIES Q: SWITCHING AND SIGNALLING

Intelligent Network

---

## **Interface Recommendation for intelligent network Capability Set 2: Part 4**

ITU-T Recommendation Q.1228 – Fascicle 4/5

(Previously CCITT Recommendation)

---

## ITU-T Q-SERIES RECOMMENDATIONS

**SWITCHING AND SIGNALLING**

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4 AND No. 5	Q.120–Q.249
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
<b>INTELLIGENT NETWORK</b>	<b>Q.1200–Q.1999</b>
BROADBAND ISDN	Q.2000–Q.2999

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

**Recommendation Q.1228**

**INTERFACE RECOMMENDATION FOR INTELLIGENT  
NETWORK CAPABILITY SET 2**

**FASCICLE 4**

## FOREWORD

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

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

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

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

## NOTE

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

## INTELLECTUAL PROPERTY RIGHTS

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

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

© ITU 1999

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

# CONTENTS

## Page

### PART 1

1	Introduction .....	1
2	General .....	1
2.1	Normative references.....	1
2.2	Abbreviations and acronyms .....	3
2.3	Conventions.....	8
3	Interface recommendation for telecommunication services.....	8
3.1	General .....	8
3.1.1	Definition methodology .....	8
3.1.2	Example physical scenarios .....	9
3.1.3	INAP protocol architecture .....	17
3.1.4	INAP addressing .....	18
3.1.5	Relationship between Recommendation Q.1224 and this Recommendation.....	19
3.1.6	Compatibility mechanisms used for INAP.....	24
3.2	SACF/MACF rules.....	25
3.2.1	Reflection of TCAP AC.....	25
3.2.2	Sequential/parallel execution of operations .....	25
4	Common IN CS-2 Types.....	25
4.1	Data types .....	25
4.2	Error types .....	53
4.3	Operations codes .....	55
4.4	Error codes .....	59
4.5	Classes .....	60
4.6	Object identifiers .....	68
5	SSF/SCF interface .....	73
5.1	Operations and arguments .....	73
5.2	SSF/SCF packages, contracts and Application Contexts .....	119
5.2.1	Protocol overview .....	119
5.2.2	SSF/SCF ASN.1 module.....	138
6	SCF/SRF interface.....	160
6.1	SCF/SRF operations and arguments.....	160
6.2	SRF/SCF contracts, packages and Application Contexts.....	165

	<b>Page</b>
6.2.1 Protocol overview .....	165
6.2.2 SRF/SCF ASN.1 modules.....	166
7 SCF-SDF interface .....	169
7.1 Introduction to the reuse of X.500 for SDF interfaces .....	169
7.1.1 Alignment between the X.500 concepts and the IN.....	169
7.1.2 Use of a limited subset of X.500.....	170
7.1.3 Working assumptions.....	170
7.2 The SDF Information Model.....	170
7.2.1 Information framework.....	170
7.2.2 Basic Access Control .....	172
7.2.3 Attribute contexts .....	173
7.2.4 Attribute definitions .....	174
7.3 The SCF-SDF Interface Protocol .....	175
7.3.1 Information types and common procedures.....	175
7.3.2 Operations .....	177
7.3.3 Errors.....	180
7.4 Protocol overview.....	181
7.4.1 Remote Operations.....	181
7.4.2 Directory ROS-Objects and Contracts .....	181
7.4.3 DAP Contract and Packages .....	182
7.5 Directory protocol abstract syntax.....	183
7.5.1 Abstract syntaxes .....	183
7.5.2 Directory application contexts .....	184
7.5.3 Operation codes.....	185
7.5.4 Error codes .....	185
7.5.5 Versions and the rules for extensibility.....	186
7.6 Conformance .....	187
7.6.1 Conformance by SCFs .....	187
7.6.2 Conformance by SDFs .....	188
7.7 ASN.1 modules for the SCF-SDF interface .....	189
7.7.1 IN-CS2-SDF-InformationFramework module.....	189
7.7.2 IN-CS2-SDF-BasicAccessControl Module.....	191
7.7.3 IN-CS2-SCF-SDF-Operations Module.....	193
7.7.4 IN-CS2-SCF-SDF-Protocol Module.....	195
8 SDF/SDF interface .....	198
8.1 Introduction to the IN X.500 DSP and DISP Subset.....	198
8.2 Working assumptions.....	198

	<b>Page</b>
8.3 The IN X.500 DISP Subset .....	199
8.3.1 Shadowing agreement specification.....	199
8.3.2 DSA Shadow Bind .....	199
8.3.3 IN-DSA Shadow Unbind .....	199
8.3.4 Coordinate Shadow Update.....	199
8.3.5 Update Shadow .....	200
8.3.6 Request Shadow Update .....	201
8.4 The IN X.500 DSP Subset.....	202
8.4.1 Information types and common procedures.....	202
8.4.2 DSA Bind.....	206
8.4.3 IN DSA Unbind.....	206
8.4.4 Chained Operations.....	206
8.4.5 Chained Errors .....	207
8.5 Protocol overview.....	207
8.5.1 ROS-Objects and contracts .....	207
8.5.2 DSP contract and packages .....	208
8.5.3 DISP contract and packages.....	209
8.6 Protocol abstract syntax.....	210
8.6.1 DSP abstract syntax.....	210
8.6.2 DISP Abstract Syntax.....	210
8.6.3 Directory System Application Context .....	211
8.6.4 Directory Shadow Application Context.....	211
8.6.5 Versions and the rules for extensibility.....	212
8.7 Conformance .....	214
8.7.1 Conformance by SDFs .....	214
8.7.2 Conformance by a shadow supplier .....	216
8.7.3 Conformance by a shadow consumer.....	216
8.8 ASN.1 modules for the SDF-SDF interface.....	217
8.8.1 IN-CS2-SDF-SDF-Protocol Module.....	217
9 SCF/SCF interface.....	221
9.1 SCF/SCF operations and arguments.....	221
9.2 SCF/SCF contracts, packages and Application Contexts.....	234
9.2.1 Protocol overview .....	234
9.2.2 ASN.1 modules .....	238
10 SCF/CUSF interface.....	243
10.1 Operations and arguments .....	243

	<b>Page</b>
10.2 SCF/CUSF Contracts, Operation Packages, and Application Contexts.....	249
10.2.1 Protocol overview .....	249
10.2.2 ASN.1 module.....	251
 <b>PART 2</b>	
11 SSF application entity procedures.....	255
11.1 General .....	255
11.2 Model and interfaces .....	255
11.3 Relations between SSF FSM and the CCF and maintenance functions.....	256
11.4 SSF management finite state model (SSME FSM).....	259
11.5 SSF switching state model (SSM) FSM.....	260
11.5.1 Finite State Model for Call Segment Association (CSA) .....	263
11.5.2 Finite State Model for Call Segment.....	268
11.6 Assisting SSF FSM .....	279
11.6.1 State aa: Idle.....	279
11.6.2 State ab: Waiting For Instructions.....	280
11.6.3 State ac: Waiting For End Of User Interaction .....	281
11.7 Handed-off SSF FSM.....	282
11.7.1 State ha: Idle.....	282
11.7.2 State hb: Waiting For Instructions .....	283
11.7.3 State hc: Waiting For End Of User Interaction.....	284
11.8 User Service Interaction USI FSM.....	285
12 SCF application entity procedures.....	286
12.1 General .....	286
12.2 Model and interfaces .....	286
12.3 Relationship between the SCF FSM and the SLPs/maintenance functions .....	287
12.4 Partial SCF Management Entity (SCME) State Transition Diagram.....	289
12.4.1 State M1: Status report idle.....	291
12.4.2 State M2: Waiting for SSF Resource Status Report .....	291
12.4.3 State M3: Service filtering idle .....	291
12.4.4 State M4: Waiting for SSF service filtering response.....	292
12.4.5 State M5: Activity test idle .....	292
12.4.6 State M6: Waiting for activity test response .....	292
12.4.7 State M7: ManageTriggerData idle.....	292
12.4.8 State M8: Waiting for ManageTriggerData activity test response.....	293
12.4.9 The Resource Control Object.....	293



	<b>Page</b>
12.5 The SCF Call State Model (SCSM) .....	293
12.5.1 SSF/SRF-related states (SCSM-SSF/SRF) .....	294
12.5.2 SDF-related states (SCSM-SDF) .....	330
12.5.3 SCF-related states.....	331
12.5.4 CUSF-related states (SCSM-CUSF) .....	342
12.5.5 USI_SCF FSM .....	346
13 SRF application entity procedures.....	346
13.1 General .....	346
13.2 Model and interfaces .....	347
13.3 Relationship between the SRF FSM and maintenance functions/bearer connection handling .....	348
13.4 The SRSM .....	349
13.4.1 State 1: Idle .....	351
13.4.2 State 2: Connected .....	352
13.4.3 State 3: User interaction .....	353
13.5 Example SRF control procedures.....	354
13.5.1 SRF connect procedures.....	355
13.5.2 SRF end user interaction procedures.....	359
13.5.3 SRF disconnection procedures.....	361
13.5.4 Examples illustrating Complete User Interaction Sequences .....	364
14 SDF application entity procedures .....	371
14.1 General .....	371
14.2 Model and interfaces .....	372
14.3 The SDF FSM structure .....	373
14.4 SDF state transition models.....	374
14.4.1 SDF state transition model for SCF-related states .....	374
14.4.2 SDF state transition model for SDF-related states .....	376
15 CUSF application entity procedures.....	394
15.1 General .....	394
15.2 Model and interfaces .....	394
15.2.1 Background for the modelling and protocol .....	395
15.2.2 Modelling and protocol.....	396
15.3 Relations between CUSF FSM and the SSF/CCF and maintenance functions.....	397
15.4 CUSF management finite state model (CUSME FSM) .....	398
15.5 CUSF state transition diagram .....	398
15.5.1 State a: Idle.....	400

	<b>Page</b>
15.5.2 State b: Waiting For Instructions .....	401
15.5.3 State c: Monitoring.....	401
16 Error procedures .....	402
16.1 Operation related error procedures.....	402
16.1.1 AttributeError.....	402
16.1.2 Cancelled.....	404
16.1.3 CancelFailed.....	405
16.1.4 DSAReferral.....	406
16.1.5 ETCFailed .....	407
16.1.6 ExecutionError .....	408
16.1.7 ImproperCallerResponse.....	409
16.1.8 MissingCustomerRecord.....	411
16.1.9 MissingParameter.....	418
16.1.10 Name Error.....	428
16.1.11 ParameterOutOfRange .....	430
16.1.12 Referral.....	432
16.1.13 RequestedInfoError .....	433
16.1.14 ScfReferral .....	433
16.1.15 Security .....	435
16.1.16 Service.....	443
16.1.17 Shadow .....	444
16.1.18 SystemFailure.....	446
16.1.19 TaskRefused.....	449
16.1.20 UnavailableResource.....	452
16.1.21 UnexpectedComponentSequence.....	453
16.1.22 UnexpectedDataValue.....	456
16.1.23 UnexpectedParameter .....	459
16.1.24 UnknownLegID.....	462
16.1.25 UnknownResource .....	463
16.1.26 Update .....	463
16.1.27 ChainingRefused.....	464
16.1.28 DirectoryBindError .....	467
16.1.29 ScfBindFailure .....	469
16.1.30 ScfTaskRefused.....	471
16.2 Entity related error procedures .....	472
16.2.1 Expiration of T <sub>SSF</sub> .....	472
16.2.2 Expiration of T <sub>SRF</sub> .....	472
16.2.3 Expiration of T <sub>cusf</sub> .....	473

**PART 3**

17	Detailed operation procedures.....	475
17.1	ActivateServiceFiltering procedure.....	475
17.1.1	General description .....	475
17.1.2	Invoking entity (SCF).....	478
17.1.3	Responding entity (SSF) .....	479
17.2	ActivationReceivedAndAuthorized procedure .....	480
17.2.1	General description .....	480
17.2.2	Invoking entity (CUSF).....	481
17.2.3	Responding entity (SCF).....	481
17.3	ActivityTest procedure.....	481
17.3.1	General description .....	481
17.3.2	Invoking entity (SCF).....	482
17.3.3	Responding entity (SSF) .....	482
17.3.4	Responding entity (CUSF).....	482
17.3.5	Responding entity (controlling SCF or supporting SCF).....	483
17.4	AddEntry procedure .....	483
17.4.1	General description .....	483
17.4.2	Invoking entity (SCF).....	483
17.4.3	Responding entity (SDF).....	484
17.5	AnalysedInformation procedure.....	485
17.5.1	General description .....	485
17.5.2	Invoking entity (SSF).....	488
17.5.3	Responding entity (SCF).....	489
17.5.4	Error handling .....	490
17.6	AnalyseInformation procedure.....	490
17.6.1	General description .....	490
17.6.2	Invoking entity (SCF).....	491
17.6.3	Responding entity (SSF) .....	492
17.7	ApplyCharging procedure .....	494
17.7.1	General description .....	494
17.7.2	Invoking entity (SCF).....	495
17.7.3	Responding entity (SSF) .....	495
17.8	ApplyChargingReport procedure .....	496
17.8.1	General description .....	496
17.8.2	Invoking entity (SSF).....	496

	<b>Page</b>
17.8.3 Responding entity (SCF).....	497
17.9 AssistRequestInstructions procedure.....	497
17.9.1 General description .....	497
17.9.2 Invoking entity (SSF/SRF).....	497
17.9.3 Responding entity (SCF).....	498
17.10 AssociationReleaseRequested procedure .....	498
17.10.1 General description .....	498
17.10.2 Invoking entity (CUSF).....	499
17.10.3 Responding entity (SCF).....	500
17.11 AuthorizeTermination procedure .....	500
17.11.1 General description .....	500
17.11.2 Invoking entity (SSF/SRF).....	501
17.11.3 Responding entity (SSF) .....	502
17.12 CallGap procedure.....	502
17.12.1 General description .....	502
17.12.2 Invoking entity (SCF).....	505
17.12.3 Responding entity (SSF) .....	505
17.13 CallInformationReport procedure .....	506
17.13.1 General description .....	506
17.13.2 Invoking entity (SSF).....	507
17.13.3 Responding entity (SCF).....	508
17.13.4 Error handling .....	508
17.14 CallInformationRequest procedure .....	508
17.14.1 General description .....	508
17.14.2 Invoking entity (SCF).....	509
17.14.3 Responding entity (SSF) .....	510
17.15 Cancel procedure.....	510
17.15.1 General description .....	510
17.15.2 Invoking entity (SCF).....	511
17.15.3 Responding entity (SRF).....	511
17.15.4 Responding entity (SSF) .....	512
17.16 CancelStatusReportRequest procedure .....	512
17.16.1 General description .....	512
17.16.2 Invoking entity (SCF).....	512
17.16.3 Responding entity (SSF) .....	512
17.17 chainedAddEntry procedure.....	513
17.17.1 General description .....	513
17.17.2 Invoking entity (SDF) .....	513

	<b>Page</b>
17.17.3 Responding entity (SDF).....	514
17.18 ChainedConfirmedNotificationProvided procedure.....	514
17.18.1 General description .....	514
17.18.2 Invoking entity (chaining initiator supporting SCF).....	515
17.18.3 Responding entity (chaining terminator supporting SCF) .....	515
17.19 ChainedConfirmedReportChargingInformation procedure.....	516
17.19.1 General description .....	516
17.19.2 Invoking entity (chaining initiator supporting SCF).....	516
17.19.3 Responding entity (chaining terminator supporting SCF) .....	516
17.20 ChainedEstablishChargingRecord procedure.....	516
17.20.1 General description .....	517
17.20.2 Invoking entity (chaining terminator supporting SCF).....	517
17.20.3 Responding entity (chaining initiator supporting SCF) .....	518
17.21 chainedExecute procedure.....	518
17.21.1 General description .....	518
17.21.2 Invoking entity (SDF) .....	518
17.21.3 Responding entity (SDF).....	519
17.22 ChainedHandlingInformationRequest procedure.....	520
17.22.1 General description .....	520
17.22.2 Invoking entity (chaining initiator supporting SCF).....	520
17.22.3 Responding entity (chaining terminator supporting SCF) .....	521
17.23 ChainedHandlingInformationResult procedure.....	521
17.23.1 General description .....	521
17.23.2 Invoking entity (chaining terminator supporting SCF).....	522
17.23.3 Responding entity (chaining initiator supporting SCF) .....	522
17.24 chainedModifyEntry procedure.....	522
17.24.1 General description .....	522
17.24.2 Invoking entity (SDF) .....	523
17.24.3 Responding entity (SDF).....	523
17.25 ChainedNetworkCapability procedure .....	524
17.25.1 General description .....	524
17.25.2 Invoking entity (chaining terminator supporting SCF).....	525
17.25.3 Responding entity (chaining initiator supporting SCF) .....	525
17.26 ChainedNotificationProvided procedure.....	525
17.26.1 General description .....	525
17.26.2 Invoking entity (chaining initiator supporting SCF).....	526
17.26.3 Responding entity (chaining terminator supporting SCF) .....	526
17.27 ChainedReportChargingInformation procedure.....	527

	<b>Page</b>
17.27.1 General description .....	527
17.27.2 Invoking entity (chaining initiator supporting SCF).....	527
17.27.3 Responding entity (chaining terminator supporting SCF) .....	528
17.28 ChainedProvideUserInfo procedure.....	528
17.28.1 General description .....	528
17.28.2 Invoking entity (chaining terminator supporting SCF).....	529
17.28.3 Responding entity (chaining initiator supporting SCF) .....	529
17.29 chainedRemoveEntry procedure.....	529
17.29.1 General description .....	529
17.29.2 Invoking entity (SDF) .....	530
17.29.3 Responding entity (SDF).....	530
17.30 ChainedRequestNotification procedure .....	531
17.30.1 General description .....	531
17.30.2 Invoking entity (chaining terminator supporting SCF).....	532
17.30.3 Responding entity (chaining initiator supporting SCF) .....	532
17.31 chainedSearch procedure.....	532
17.31.1 General description .....	532
17.31.2 Invoking entity (SDF) .....	532
17.31.3 Responding entity (SDF).....	533
17.32 CollectedInformation procedure.....	534
17.32.1 General description .....	534
17.32.2 Invoking entity (SSF).....	535
17.32.3 Responding entity (SCF).....	537
17.33 CollectInformation procedure .....	538
17.33.1 General description .....	538
17.33.2 Invoking entity (SCF).....	539
17.33.3 Responding entity (SSF) .....	539
17.34 ComponentReceived procedure .....	540
17.34.1 General description .....	540
17.34.2 Invoking entity (CUSF).....	541
17.34.3 Responding entity (SCF).....	541
17.35 ConfirmedNotificationProvided procedure.....	542
17.35.1 General description .....	542
17.35.2 Invoking entity (controlling SCF).....	542
17.35.3 Responding entity (supporting SCF).....	543
17.36 ConfirmedReportChargingInformation procedure.....	543
17.36.1 General description .....	543
17.36.2 Invoking entity (controlling SCF).....	544

	<b>Page</b>
17.36.3 Responding entity (supporting SCF).....	544
17.37 Connect procedure.....	545
17.37.1 General description .....	545
17.37.2 Invoking entity (SCF).....	547
17.37.3 Responding entity (SSF) .....	548
17.38 ConnectToResource procedure .....	549
17.38.1 General description .....	549
17.38.2 Invoking entity (SCF).....	550
17.38.3 Responding entity (SSF) .....	550
17.39 Continue procedure .....	551
17.39.1 General description .....	551
17.39.2 Invoking entity (SCF).....	551
17.39.3 Responding entity (SSF) .....	551
17.40 ContinueWithArgument procedure .....	552
17.40.1 General description .....	552
17.40.2 Invoking entity (SCF).....	552
17.40.3 Responding entity (SSF) .....	553
17.41 CoordinateShadowUpdate procedure.....	553
17.41.1 General description .....	553
17.41.2 Supplier entity (SDF) .....	554
17.41.3 Consumer entity (SDF) .....	555
17.42 CreateCallSegmentAssociation procedure .....	556
17.42.1 General description .....	556
17.42.2 Invoking entity (SCF).....	556
17.42.3 Responding entity (SSF) .....	556
17.43 in-directoryBind procedure.....	557
17.43.1 General description .....	557
17.43.2 Invoking entity (SCF).....	557
17.43.3 Responding entity (SDF).....	557
17.44 DirectoryUnbind procedure.....	558
17.44.1 General description .....	558
17.44.2 Invoking entity (SCF).....	558
17.44.3 Responding entity (SDF).....	558
17.45 DisconnectForwardConnection procedure .....	559
17.45.1 General description .....	559
17.45.2 Invoking entity (SCF).....	559
17.45.3 Responding entity (SSF) .....	560
17.46 DisconnectForwardConnectionWithArgument procedure.....	560

	<b>Page</b>
17.46.1 General description .....	560
17.46.2 Invoking entity (SCF).....	561
17.46.3 Responding entity (SSF) .....	561
17.47 DisconnectLeg procedure.....	562
17.47.1 General description .....	562
17.47.2 Invoking entity (SCF).....	562
17.47.3 Responding entity (SSF) .....	562
17.48 dSABind procedure .....	563
17.48.1 General description .....	563
17.48.2 Invoking entity (SDF) .....	563
17.48.3 Responding entity (SDF).....	564
17.49 DSAShadowBind procedure .....	564
17.49.1 General description .....	564
17.49.2 Supplier entity (SDF) .....	565
17.49.3 Consumer entity (SDF) .....	567
17.50 in-DSAShadowUnbind procedure.....	568
17.50.1 General description .....	568
17.50.2 Supplier entity (SDF) .....	569
17.50.3 Consumer entity (SDF) .....	569
17.51 EntityReleased procedure.....	570
17.51.1 General description .....	570
17.51.2 Invoking entity (SSF).....	571
17.51.3 Responding entity (SCF).....	571
17.52 EstablishChargingRecord procedure.....	572
17.52.1 General description .....	572
17.52.2 Invoking entity (supporting SCF).....	572
17.52.3 Responding entity (controlling SCF) .....	573
17.53 EstablishTemporaryConnection procedure .....	573
17.53.1 General description .....	573
17.53.2 Invoking entity (SCF).....	574
17.53.3 Responding entity (SSF) .....	574
17.54 EventNotificationCharging procedure.....	575
17.54.1 General description .....	575
17.54.2 Invoking entity (SSF).....	576
17.54.3 Responding entity (SCF).....	576
17.55 EventReportBCSM procedure.....	577
17.55.1 General description .....	577
17.55.2 Invoking entity (SSF).....	579



	<b>Page</b>
17.55.3 Responding entity (SCF).....	579
17.56 EventReportFacility procedure.....	580
17.56.1 General description .....	580
17.56.2 Invoking entity (SSF).....	580
17.56.3 Responding entity (SCF).....	581
17.57 Execute procedure .....	581
17.57.1 General description .....	581
17.57.2 Invoking entity (SCF).....	582
17.57.3 Responding entity (SDF).....	582
17.58 FacilitySelectedAndAvailable procedure.....	584
17.58.1 General description .....	584
17.58.2 Invoking entity (SSF).....	585
17.58.3 Responding entity (SCF).....	585
17.59 FurnishChargingInformation procedure.....	586
17.59.1 General description .....	586
17.59.2 Invoking entity (SCF).....	586
17.59.3 Responding entity (SCF).....	587
17.60 HandlingInformationRequest procedure .....	588
17.60.1 General description .....	588
17.60.2 Invoking entity (controlling SCF).....	589
17.60.3 Responding entity (supporting SCF).....	590
17.61 HandlingInformationResult procedure.....	591
17.61.1 General description .....	591
17.61.2 Invoking entity (supporting SCF).....	592
17.61.3 Responding entity (controlling SCF) .....	592
17.62 HoldCallInNetwork procedure .....	593
17.62.1 General description .....	593
17.62.2 Invoking entity (SCF).....	593
17.62.3 Responding entity (SSF) .....	593
17.63 in-DSAUnbind procedure.....	593
17.63.1 General description .....	593
17.63.2 Invoking entity (SDF) .....	594
17.63.3 Responding entity (SDF).....	594
17.64 InitialDP procedure .....	594
17.64.1 General description .....	594
17.64.2 Invoking entity (SSF).....	598
17.64.3 Responding entity (SCF).....	599
17.65 InitiateAssociation procedure.....	599

	<b>Page</b>
17.65.1 General description .....	599
17.65.2 Invoking entity (SCF).....	599
17.65.3 Responding entity (CUSF).....	600
17.66 InitiateCallAttempt procedure.....	600
17.66.1 General description .....	600
17.66.2 Invoking entity (SCF).....	601
17.66.3 Responding entity (SSF) .....	602
17.67 ManageTriggerData procedure.....	602
17.67.1 General description .....	602
17.67.2 Invoking entity (SCF).....	603
17.67.3 Responding entity (SSF) .....	603
17.68 MergeCallSegments procedure .....	604
17.68.1 General description .....	604
17.68.2 Invoking entity (SCF).....	604
17.68.3 Responding entity (SSF) .....	605
17.69 ModifyEntry procedure .....	605
17.69.1 General description .....	605
17.69.2 Invoking entity (SCF).....	605
17.69.3 Responding entity (SDF).....	606
17.70 MoveCallSegments procedure .....	607
17.70.1 General description .....	607
17.70.2 Invoking entity (SCF).....	608
17.70.3 Responding entity (SSF) .....	608
17.71 MoveLeg procedure.....	608
17.71.1 General description .....	608
17.71.2 Invoking entity (SCF).....	609
17.71.3 Responding entity (SSF) .....	609
17.72 NetworkCapability procedure .....	610
17.72.1 General description .....	610
17.72.2 Invoking entity (supporting SCF).....	610
17.72.3 Responding entity (controlling SCF) .....	611
17.73 NotificationProvided procedure .....	611
17.73.1 General description .....	611
17.73.2 Invoking entity (controlling SCF).....	612
17.73.3 Responding entity (supporting SCF).....	612
17.74 OAbandon procedure .....	613
17.74.1 General description .....	613
17.74.2 Invoking entity (SSF).....	613

	<b>Page</b>
17.74.3 Responding entity (SCF).....	613
17.75 OAnswer procedure.....	614
17.75.1 General description .....	614
17.75.2 Invoking entity (SSF).....	615
17.75.3 Responding entity (SCF).....	615
17.76 OCalledPartyBusy procedure .....	616
17.76.1 General description .....	616
17.76.2 Invoking entity (SSF).....	617
17.76.3 Responding entity (SCF).....	618
17.77 ODisconnect procedure .....	618
17.77.1 General description .....	618
17.77.2 Invoking entity (SSF).....	619
17.77.3 Responding entity (SCF).....	620
17.78 OMidCall procedure.....	620
17.78.1 General description .....	620
17.78.2 Invoking entity (SSF).....	621
17.78.3 Responding entity (SCF).....	622
17.79 ONoAnswer procedure.....	622
17.79.1 General description .....	622
17.79.2 Invoking entity (SSF).....	623
17.79.3 Responding entity (SCF).....	624
17.80 OriginationAttempt procedure .....	625
17.80.1 General description .....	625
17.80.2 Invoking entity (SSF).....	626
17.80.3 Responding entity (SCF).....	626
17.81 OriginationAttemptAuthorized procedure .....	626
17.81.1 General description .....	626
17.81.2 Invoking entity (SSF).....	627
17.81.3 Responding entity (SCF).....	628
17.82 OSuspended procedure.....	628
17.82.1 General description .....	628
17.82.2 Invoking entity (SSF).....	629
17.82.3 Responding entity (SCF).....	629
17.83 PlayAnnouncement procedure.....	630
17.83.1 General description .....	630
17.83.2 Invoking entity (SCF).....	631
17.83.3 Responding entity (SRF).....	632
17.84 PromptAndCollectUserInformation procedure .....	632

	<b>Page</b>
17.84.1 General description .....	632
17.84.2 Invoking entity (SCF).....	636
17.84.3 Responding entity (SRF).....	637
17.85 PromptAndReceiveMessage procedure.....	638
17.85.1 General description .....	638
17.85.2 Invoking entity (SCF).....	641
17.85.3 Responding entity (SRF).....	641
17.86 ProvideUserInformation procedure.....	642
17.86.1 General description .....	642
17.86.2 Invoking entity (supporting SCF).....	643
17.86.3 Responding entity (controlling SCF) .....	643
17.87 Reconnect procedure .....	644
17.87.1 General description .....	644
17.87.2 Invoking entity (SCF).....	645
17.87.3 Responding entity (SSF) .....	645
17.88 ReleaseAssociation procedure.....	645
17.88.1 General description .....	645
17.88.2 Invoking entity (SCF).....	646
17.88.3 Responding entity (CUSF).....	646
17.89 ReleaseCall procedure.....	646
17.89.1 General description .....	646
17.89.2 Invoking entity (SCF).....	647
17.89.3 Responding entity (SSF) .....	647
17.90 RemoveEntry procedure.....	648
17.90.1 General description .....	648
17.90.2 Invoking entity (SCF).....	648
17.90.3 Responding entity (SDF).....	648
17.91 ReportChargingInformation procedure .....	649
17.91.1 General description .....	649
17.91.2 Invoking entity (controlling SCF).....	650
17.91.3 Responding entity (supporting SCF).....	650
17.92 ReportUTSI procedure .....	650
17.92.1 General description .....	650
17.92.2 Invoking entity (SSF).....	651
17.92.3 Responding entity (SCF).....	651
17.93 RequestCurrentStatusReport procedure .....	652
17.93.1 General description .....	652
17.93.2 Invoking entity (SCF).....	652

	<b>Page</b>
17.93.3 Responding entity (SSF) .....	652
17.94 RequestEveryStatusChangeReport procedure.....	653
17.94.1 General description .....	653
17.94.2 Invoking entity (SCF).....	653
17.94.3 Responding entity (SSF) .....	654
17.95 RequestFirstStatusMatchReport procedure.....	654
17.95.1 General description .....	654
17.95.2 Invoking entity (SCF).....	655
17.95.3 Responding entity (SSF) .....	655
17.96 RequestNotification procedure.....	656
17.96.1 General description .....	656
17.96.2 Invoking entity (supporting SCF).....	656
17.96.3 Responding entity (controlling SCF) .....	657
17.97 RequestNotificationChargingEvent procedure.....	657
17.97.1 General description .....	657
17.97.2 Invoking entity (SCF).....	658
17.97.3 Responding entity (SSF) .....	658
17.98 RequestReportBCSMEEvent procedure.....	659
17.98.1 General description .....	659
17.98.2 Invoking entity (SCF).....	661
17.98.3 Responding entity (SSF) .....	661
17.99 RequestReportBCUSMEEvent procedure.....	662
17.99.1 General description .....	662
17.99.2 Parameters .....	662
17.99.3 Invoking entity (SCF).....	662
17.99.4 Responding entity (CUSF).....	662
17.100 RequestReportFacilityEvent procedure.....	663
17.100.1 General description .....	663
17.100.2 Invoking entity (SCF) .....	663
17.100.3 Responding entity (SSF).....	664
17.101 RequestReportUTSI procedure .....	664
17.101.1 General description .....	664
17.101.2 Invoking entity (SCF) .....	665
17.101.3 Responding entity (SSF).....	665
17.102 RequestShadowUpdate procedure.....	665
17.102.1 General description .....	665
17.102.2 Supplier entity (SDF).....	666
17.102.3 Consumer entity (SDF).....	667

	<b>Page</b>
17.103 ResetTimer procedure .....	667
17.103.1 General description .....	667
17.103.2 Invoking entity (SCF) .....	668
17.103.3 Responding entity (SSF) .....	668
17.104 RouteSelectFailure procedure .....	669
17.104.1 General description .....	669
17.104.2 Invoking entity (SSF) .....	669
17.104.3 Responding entity (SCF) .....	670
17.105 SCFBind procedure .....	671
17.105.1 General description .....	671
17.105.2 Responding entity (supporting SCF) .....	672
17.106 scfBind procedure (in the chaining case) .....	672
17.106.1 General description .....	672
17.106.2 Invoking entity (chaining initiator supporting SCF) .....	673
17.106.3 Responding entity (chaining terminator supporting SCF) .....	673
17.107 SCFUnBind procedure .....	673
17.107.1 General description .....	673
17.107.2 Invoking entity (controlling SCF) .....	673
17.107.3 Responding entity (supporting SCF) .....	674
17.108 scfUnBind procedure (in the chaining case) .....	674
17.108.1 General description .....	674
17.108.2 Invoking entity (chaining terminator supporting SCF) .....	674
17.108.3 Responding entity (chaining terminator supporting SCF) .....	675
17.109 ScriptClose procedure .....	675
17.109.1 General description .....	675
17.109.2 Invoking entity (SCF) .....	675
17.109.3 Responding entity (SRF) .....	676
17.110 ScriptEvent procedure .....	676
17.110.1 General Description .....	676
17.110.2 Invoking entity (SRF) .....	676
17.110.3 Responding entity (SCF) .....	677
17.111 ScriptInformation procedure .....	678
17.111.1 General description .....	678
17.111.2 Invoking entity (SCF) .....	678
17.111.3 Responding entity (controlling SRF) .....	679
17.112 ScriptRun procedure .....	679
17.112.1 General description .....	679
17.112.2 Invoking entity (SCF) .....	680

	<b>Page</b>
17.112.3 Responding entity (SRF).....	680
17.113 Search procedure .....	680
17.113.1 General description .....	680
17.113.2 Invoking entity (SCF) .....	681
17.113.3 Responding entity (SDF) .....	681
17.114 SelectFacility procedure .....	682
17.114.1 General description .....	682
17.114.2 Invoking entity (SCF) .....	683
17.114.3 Responding entity (SSF).....	684
17.115 SelectRoute procedure.....	684
17.115.1 General description .....	684
17.115.2 Invoking entity (SCF) .....	686
17.115.3 Responding entity (SSF).....	686
17.116 SendChargingInformation procedure .....	688
17.116.1 General description .....	688
17.116.2 Invoking entity (SCF) .....	688
17.116.3 Responding entity (SSF).....	689
17.117 SendComponent procedure .....	690
17.117.1 General description .....	690
17.117.2 Invoking entity (SCF) .....	691
17.117.3 Responding entity (CUSF).....	691
17.118 SendFacilityInformation procedure.....	691
17.118.1 General description .....	691
17.118.2 Invoking entity (SCF) .....	692
17.118.3 Responding entity (SSF).....	692
17.119 SendSTUI procedure .....	693
17.119.1 General description .....	693
17.119.2 Invoking entity (SCF) .....	693
17.120 ServiceFilteringResponse procedure.....	694
17.120.1 General description .....	694
17.120.2 Invoking entity (SSF).....	694
17.120.3 Responding entity (SCF).....	695
17.121 SpecializedResourceReport procedure.....	696
17.121.1 General description .....	696
17.121.2 Invoking entity (SRF) .....	696
17.121.3 Responding entity (SCF).....	696
17.122 SplitLeg procedure .....	696
17.122.1 General description .....	696

	<b>Page</b>
17.122.2 Invoking entity (SCF) .....	697
17.122.3 Responding entity (SSF) .....	697
17.123 StatusReport procedure .....	698
17.123.1 General description .....	698
17.123.2 Invoking entity (SSF) .....	698
17.123.3 Responding entity (SCF) .....	699
17.124 TAnswer procedure .....	699
17.124.1 General description .....	699
17.124.2 Invoking entity (SSF) .....	700
17.124.3 Responding entity (SCF) .....	700
17.125 TBusy procedure .....	701
17.125.1 General description .....	701
17.125.2 Invoking entity (SSF) .....	702
17.125.3 Responding entity (SCF) .....	702
17.126 TDisconnect procedure .....	703
17.126.1 General description .....	703
17.126.2 Invoking entity (SSF) .....	704
17.126.3 Responding entity (SCF) .....	704
17.127 TerminationAttempt procedure .....	705
17.127.1 General description .....	705
17.127.2 Invoking entity (SSF) .....	706
17.127.3 Responding entity (SCF) .....	706
17.128 TermAttemptAuthorized procedure .....	706
17.128.1 General description .....	706
17.128.2 Invoking entity (SSF) .....	707
17.128.3 Responding entity (SCF) .....	708
17.129 TMidCall procedure .....	708
17.129.1 General description .....	708
17.129.2 Invoking entity (SSF) .....	709
17.129.3 Responding entity (SCF) .....	710
17.130 TNoAnswer procedure .....	710
17.130.1 General description .....	710
17.130.2 Invoking entity (SSF) .....	711
17.130.3 Responding entity (SCF) .....	712
17.131 TSuspended procedure .....	713
17.131.1 General description .....	713
17.131.2 Invoking entity .....	713
17.131.3 Responding entity (SCF) .....	713



	<b>Page</b>
17.132 UpdateShadow procedure.....	714
17.132.1 General description.....	714
17.132.2 Supplier entity (SDF).....	714
17.132.3 Consumer entity (SDF).....	716
 <b>PART 4</b>	
18 Services assumed from Lower Layers.....	719
18.1 Services assumed from TCAP.....	719
18.1.1 Common procedures .....	719
18.1.2 SSF-SCF interface.....	730
18.1.3 SCF-SRF interface .....	737
18.1.4 SCF-CUSF interface .....	738
18.1.5 SCF-SCF interface .....	740
18.1.6 SCF-SDF interface.....	743
18.1.7 SDF-SDF interface.....	746
18.2 Services assumed from SCCP .....	749
18.2.1 Normal procedures .....	749
18.2.2 Service functions from SCCP .....	749
19 IN generic interface security.....	752
19.1 Interface security requirements.....	752
19.1.1 Data confidentiality.....	752
19.1.2 Data integrity and data origin authentication .....	752
19.1.3 Key management.....	753
19.2 Procedures and algorithms .....	753
19.2.1 Authentication procedures .....	753
19.2.2 SPKM algorithms and negotiation.....	754
19.2.3 Three-way mutual authentication.....	755
19.2.4 Assignment of credentials.....	755
19.3 Mapping of security information flow definitions to tokens.....	755
19.4 Security FSM definitions .....	756
19.4.1 Two-way mutual authentication FSMs .....	756
19.4.2 Three-way mutual authentication FSMs .....	759

**PART 5**

Annex A.1 – Introduction to the INAP CS-1 and CS-2 SDL models .....	763
A.1.1 Introduction.....	763
A.1.2 Example for the interworking of the SSF/CCF SDL processes.....	768
A.1.3 Example for the Three-Party Call setup as seen from the environment.....	770
Annex A.2 – Transition diagrams.....	773
A.2.1 Call Segment Association transition diagram.....	773
A.2.2 Call Segment transition diagram.....	774
Annex A.3 – SDL Specification of CS-1 SSF/CCF .....	787

**PART 6**

Annex A.4 – SDL Specification of CS-2 extensions to SSF/CCF .....	788
---	-----

**PART 7**

Annex A.5 – SDL Specification of CS-2 SRF .....	789
Annex A.6 – SDL Specification of CS-2 Assist/Hand-off SSF .....	789
Annex A.7 – SDL Specification of CS-2 CUSF .....	789
Annex A.8 – SDL Specification of CS-2 SCF .....	789
Appendix I – Expanded ASN.1 source.....	790
Appendix II – Data modelling .....	819
II.1 Introduction .....	819
II.1.1 Purpose and scope.....	819
II.1.2 Assumptions.....	819
II.2 Directory Information Tree (DIT) schema.....	820
II.2.1 X.500 DIT .....	820
II.2.2 Object classes.....	822
II.2.3 Attribute types.....	823
II.2.4 DIT structure definition.....	824
Appendix III – Examples of SPKM algorithms for IN CS-2.....	826
III.1 General .....	826
III.2 Integrity Algorithm (I-ALG).....	826
III.2.1 Example–1 .....	826
III.2.2 Example–2 .....	827
III.2.3 Example–3 .....	827
III.2.4 Example–4 .....	827

	<b>Page</b>
III.3 Confidentiality Algorithm (C-ALG) .....	828
III.3.1 Example–1 .....	828
III.4 Key Establishment Algorithm (K-ALG).....	828
III.4.1 Example–1 .....	828
III.4.2 Example–2 .....	828
III.4.3 Example–3 .....	828
III.5 One-Way Function (O-ALG) for Subkey Derivation Algorithm.....	828
III.5.1 Example–1 .....	829



## Recommendation Q.1228

### INTERFACE RECOMMENDATION FOR INTELLIGENT NETWORK CAPABILITY SET 2

(Geneva, 1997)

#### PART 4

## 18 Services assumed from Lower Layers

### 18.1 Services assumed from TCAP

The SS7 application layer protocol defined in this Recommendation is a protocol to provide communication between a pair of application processes. In the SS7, environment this is represented as communication between a pair of Application Entities (AEs) using the Transaction Capabilities. The function of an AE is provided by a set of Application Service Elements (ASEs). The interaction between AEs is described in terms of their use of the services provided by the ASEs.

If Application Contexts (ACs) are to be used for FE differentiation within a physical node, then the version of TC used must support the dialogue portion of TC (i.e. *White Book*, TC).

This requirement applies to all interfaces, not just those used for internetworking.

Table 18-1 defines which versions of TC are the minimum versions required to support the defined IN interfaces:

**Table 18-1/Q.1228 – Minimum TC Recommendations  
requirements for INAP interfaces**

Interface	IN CS-2
SSF-SCF	<i>Blue Book</i> (Note)
SCF-SRF	<i>Blue Book</i> (Note)
SCF-SDF	<i>White Book</i> (1993)
SCF-SCF	<i>White Book</i> (1993)
SDF-SDF	<i>White Book</i> (1993)
CUSF-SCF	<i>Blue Book</i> (Note)
NOTE – If AC negotiation is required, then <i>White Book</i> (1993) is the minimum version required.	

#### 18.1.1 Common procedures

This subclause defines the procedures and mapping which apply between INAP and TC to be used in the absence of specific procedures and mapping instructions for the specific INAP interfaces as defined in subsequent subclauses.

#### 18.1.1.1 Normal procedures

This subclause describes the procedures and TCAP primitives that shall be used for transmitting messages between AEs under normal operation.

The INAP, as TC-user, uses only the structured dialogue facility provided by TCAP. The following situations can occur when a message is sent between two physical entities:

- a dialogue shall be established: the TC-user issues a TC-BEGIN request primitive;
- a dialogue shall be maintained: the TC-user issues a TC-CONTINUE request primitive;
- a dialogue shall no longer be maintained: the TC-user issues a TC-END request primitive with either basic end or with prearranged end depending on the following conditions:
  - Basic end:
    - In the case the dialogue is established, operations, leading to a termination of the relationship, can be transmitted by the FE with a TC-END request primitive (basic) in case the FE is not interested in the reception of any ERROR or REJECT components for these sent operations. Once the FE dialogue resources have been released, any ERROR or REJECT components received for these operations will be discarded by TC as described in Recommendation Q.774.
    - In the case where a dialogue has been established and the FE receives an operation leading to the termination of the relationship, and there is no interest by the FE to continue the dialogue, and there is no operation to be sent, a TC-END request primitive (basic) with zero components can be sent from the FE.
  - Prearranged end:

In this case, an entity is interested in possible ERROR or REJECT messages on response to sent operations leading to a termination of the relationship, the dialogue is ended with a TC-END request primitive (prearranged end) after the last associated operation timer expires. The receiving entity can end the dialogue with a TC-END request primitive (prearranged end) after successful processing of these operations (i.e. the relationship is terminated).
- in general, the use of prearranged end shall be limited to the case for both communicating entities clearly recognisable that peer entity applies prearranged end. In all other cases, basic end shall be used;
- a dialogue shall not be established: for class 2 or 4 operations only the sending TC-user issues a TC-BEGIN request primitive and ends the dialogue locally after operation time-out by means of a prearranged end. Upon reception of the TC-BEGIN indication primitive, the receiving TC-user shall end the dialogue locally.

#### 18.1.1.2 Abnormal procedures

This subclause describes the procedures and TCAP primitives that shall be used for reporting abnormal situations between AEs. The error cases are defined in clause 16.

The following primitives shall be used to report abnormal situations:

- operation errors, as defined in the INAP, are reported with TC-U-ERROR request primitive;
- rejection of a TCAP component by the TC-user shall be reported with TC-U-REJECT request primitive;
- when the FE detecting error or rejecting operation decides the termination of TC dialogue, TC-END request primitive (basic) with error or reject can be used for the termination of TC dialogue;

- when the SSF, the SRF, or the CUSF detecting error or rejecting operation recognises the possibility to continue dialogue, TC-CONTINUE request primitive with error or reject can be used for the continuation of TC dialogue;
- a dialogue shall be aborted by the TC-user with a TC-U-ABORT request primitive;
- on expiration of application timer  $T_{SSF}$  or  $T_{SRF}$  or  $T_{CUSF}$  dialogue shall be terminated by means of a TC-U-ABORT primitive with an Abort reason, regardless of whether TCAP dialogue is established or not.

For abnormal situations detected by TCAP, the same rules shall apply for reception of TC-R-REJECT indication as for transmission of TC-U-REJECT request and for transmission of TC-P-ABORT indication as for transmission of TC-U-ABORT request primitive.

The following rules shall be applied to terminate the TCAP dialogue under abnormal situations:

- in the case that abort condition is detected and TCAP dialogue is established, TCAP dialogue is terminated by TC-U-ABORT primitive with an Abort reason;
- in the case that abort condition is detected and TCAP dialogue is not established, TCAP dialogue is locally terminated by TC-U-ABORT primitive (in the case such as application time-out).

In error situations, prearranged end shall not be used to terminate the TCAP dialogue. In case any application entity encounters an error situation the peer entity shall be explicitly notified of the error, if possible. If from any entity's point of view the error encountered requires the relationship to be ended, it shall close the dialogue via a TC-END request primitive with basic end or via a TC-U-ABORT request primitive, depending on whether any pending ERROR or REJECT component is to be sent or not.

In case an entity receives a TC-END indication primitive and after all components have been considered, the FSM is not in a state to terminate the relationship, an appropriate internal error should be provided.

In cases when a dialogue needs to be closed by the initiating entity before its establishment has been completed (before the first TC indication primitive to the TC-BEGIN request primitive has been received from the responding entity), the TC-user shall issue a TC-END request primitive with prearranged end or a TC-U-ABORT request primitive. The result of these primitives will be only local, any subsequent TC indication received for this dialogue will be handled according to the abnormal procedures as specified in Recommendation Q.774.

### **18.1.1.3 Dialogue handling**

#### **18.1.1.3.1 Dialogue establishment**

The establishment of an INAP dialogue involves two application processes as described in 3.1.3, one that is the dialogue-initiator and one that is the dialogue-responder.

Application context negotiation may not be supported in all physical entities and/or all networks.

This procedure is driven by the following signals:

- a TC-BEGIN request primitive from the dialogue-initiator;
- a TC-BEGIN indication primitive occurring at the responding side;
- the first TC-CONTINUE indication primitive occurring at the initiating side or under specific conditions:
  - a TC-END indication primitive occurring at the initiating side;
  - a TC-U-ABORT indication primitive occurring at the initiating side;

- a TC-P-ABORT indication primitive occurring at the initiating side.

### **Sending of a TC-BEGIN request**

Before issuing a TC-BEGIN request primitive, SACF shall store the AC-name and, if present, the user-information parameter.

SACF shall request the invocation of the associated operations using the TC-INVOKE service. See 18.1.1.4.1 for a description of the invocation procedure.

After processing of the last invocation request, SACF shall issue a TC-BEGIN request primitive.

The initiator SACF then waits for a TC indication primitive and will not issue any other requests, except a TC-U-ABORT request or a TC-END request with the release method parameter set to "prearranged release".

If no TC indication primitive is expected because no dialogue is to be established according to the rules as stated in 18.1.2.1.1 and 18.1.2.1.2, SACF will wait for the last associated TCAP operation timer to expire and issue a TC-END request with the release method parameter set to "prearranged release".

### **Receipt of a TC-BEGIN indication**

On receipt of a TC-BEGIN indication primitive, responder SACF shall:

- Analyse the application-context-name if included in the primitive. If it is supported, process any other indication primitives received from TC as described in 18.1.1.4.1.
- If no dialogue is to be established according to the rules as stated in 18.1.2.1.1 and 18.1.2.1.2, SACF will wait for the last indication primitive from TC and issue a TC-END request with the release method parameter set to "prearranged release".
- If the application-context-name included in the primitive is not supported, issue a TC-U-ABORT request primitive. If an alternative application-context can be offered, its name is included in the TC-U-ABORT request primitive.

It is for further study whether or not the application-context-negotiation is limited only for using the TC-U ABORT primitive.

### **Receipt of the first TC-CONTINUE indication**

On receipt of the first TC-CONTINUE indication primitive for a dialogue, SACF shall check the value of the application-context-name parameter. If this value matches the one used in the TC-BEGIN request primitive, SACF shall process the following TC component handling indication primitives as described in 18.1.1.4.1; otherwise, it shall issue a TC-U-ABORT request primitive.

It is for further study whether or not the application-context-negotiation is limited only for using the TC-U ABORT primitive.

### **Receipt of a TC-END indication**

On receipt of a TC-END indication primitive in the dialogue initiated state, SACF shall check the value of the application-context-name parameter. If this value matches the one used in the TC-BEGIN request primitive, then the SACF shall process the following TC component handling indication primitives as described in 18.1.1.4.1.

### **Receipt of a TC-U-ABORT indication**

Receipt of a TC-U-ABORT indication primitive is described as part of user abort procedure (see 18.1.1.3.4). If the abort reason is application context name not supported, the responding side may propose an alternative application context name in the TC-U-ABORT indication. If an alternative



application context is proposed, the receiving entity shall check this name and if it can be supported a new dialogue may be established.

### **Receipt of a TC-P-ABORT indication**

Receipt of a TC-P-ABORT indication primitive is described as part of provider abort procedure (see 18.1.1.3.5).

### **18.1.1.3.2 Dialogue continuation**

Once established, the dialogue is said to be in a continuation phase.

Both application processes can request the transfer of INAP APDUs until one of them requests the termination of the dialogue.

### **Sending entity**

SACF shall process any component handling request primitives as described in 18.1.1.4.1.

After processing the last component handling request primitive, SACF shall issue a TC-CONTINUE request primitive.

### **Receiving entity**

On receipt of a TC-CONTINUE indication primitive SACF shall accept zero, one or several TC component handling indication primitives and process them as described in 18.1.1.4.1.

### **18.1.1.3.3 Dialogue termination**

Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it has been established when no dialogue is to be established or when a dialogue is no longer to be maintained according to the rules as stated in 18.1.2.1.1 and 18.1.2.1.2.

The dialogue termination procedure is driven by the following events:

- a TC-END request primitive;
- a TC-END indication primitive.

### **Sending of TC-END request**

When the dialogue shall no longer be maintained, SACF shall process any component handling request primitives as described in 18.1.1.4.1.

After processing the last component handling request primitive (if any), SACF shall issue a TC-END request primitive with the release method parameter set to "basic end" or "prearranged release", according to the rules as stated in 18.1.2.1.1 and 18.1.2.1.2.

When no dialogue is to be established, refer to 18.1.1.3.1.

### **Receipt of a TC-END indication**

On receipt of a TC-END indication primitive, the SACF shall accept any component handling indication primitives and process them as described in 18.1.1.4.1.

After processing the last component handling primitive, all dialogue related resources are released.

#### **18.1.1.3.4 User Abort**

Both the dialogue-initiator and the dialogue-responder have the ability to abort a dialogue at any time.

The user abort procedure is driven by one of the following events:

- a TC-U-ABORT request primitive;
- a TC-U-ABORT indication primitive.

#### **Sending of TC-U-ABORT request**

After issuing a TC-U-ABORT request primitive, all dialogue related resources are released.

#### **Receipt of a TC-U-ABORT indication**

On receipt of a TC-U-ABORT indication, all dialogue related resources are released.

#### **18.1.1.3.5 Provider Abort**

TC has the ability to abort a dialogue at both the dialogue-initiator side and the dialogue-responder side.

The provider abort procedure is driven by the following event:

- a TC-P-ABORT indication primitive.

#### **Receipt of a TC-P-ABORT indication**

On receipt of a TC-P-ABORT indication, all dialogue related resources are released.

#### **18.1.1.3.6 Mapping to TC dialogue primitives**

The TC-UNI service is not used by INAP.

The mapping of parameters onto the TC Dialogue services is as follows:

The use of parameters of the TC-BEGIN service is as defined in 18.1.1.3.7 with the following qualifications:

- The Destination Address parameter of the TC-BEGIN service shall be set to the INAP address of the AE which is to respond to the TC-BEGIN service.  
  
NOTE 1 – The address used in this parameter may be mapped by SCCP address translation to one of a number of alternative AEs.
- The Application Context Name parameter of the TC-BEGIN service shall be set according to the specific interface being used between the initiating AE and the responding AE.
- The Originating Address parameter of the TC-BEGIN service shall be set to the unambiguous INAP address of the AE initiating the TC-BEGIN service.

The use of parameters of the TC-CONTINUE service is as defined in 18.1.1.3.7 with the following qualifications:

- The Application Context Name parameter of the TC-CONTINUE service shall be set to the value of the Application Context Name parameter of the TC-BEGIN service for the same Dialogue ID parameter value.
- If present, the Originating Address parameter of the TC-CONTINUE service shall be set to the unambiguous INAP address of the AE initiating the TC-CONTINUE service. This parameter is only present in the first TC-CONTINUE service after a TC-BEGIN service with the same Dialogue ID parameter value.

The use of parameters of the TC-END service is as defined in 18.1.1.3.7 with the following qualifications:

- The Application Context Name parameter of the TC-END service shall be set to the value of the Application Context Name parameter of the TC-BEGIN service for the same Dialogue ID parameter value. This parameter is only present if the TC-END service is used immediately after the TC-BEGIN service.

The use of parameters of the TC-U-ABORT service is as defined in 18.1.1.3.7 with the following qualifications:

- The Abort Reason parameter of the TC-U-ABORT service shall be used as specified in Recommendation Q.771.
- The Application Context Name parameter of the TC-U-ABORT service shall be set to either the value used in the TC-BEGIN service or an alternative value which can be used to establish the dialogue between the initiating AE and the responding AE.

NOTE 2 – This parameter is only present if the TC-U-ABORT is the immediate response to a TC-BEGIN indication.

The use of parameters of the TC-P-ABORT service is as defined in 18.1.1.3.7 with the following qualifications:

- The P-Abort parameter of the TC-P-ABORT service is set by TC to indicate the reason why TC aborted the dialogue. It shall take the values as defined in Recommendation Q.771.

### **18.1.1.3.7 Default mapping to TC dialogue parameters**

#### **Dialogue Id**

The value of this parameter is associated with the INAP invocation in an implementation-dependent manner. This parameter uniquely identifies a specific TC dialogue to a remote INAP AE for an INAP AE.

#### **Application-context-name**

The application-context-name parameter is set according to the set of operations which need to be supported by the TC dialogue. The defined Application Context Names can be found in clauses 5 to 11.

#### **User information**

This parameter may be used by both initiating and responding application processes in a network-operator specific manner.

#### **Component present**

This parameter is used by SACF as described in Recommendation Q.771.

#### **Termination**

The value of the release method parameter of the TC-END request primitive is set by SACF according to the rules as stated in 18.1.2.1.1 and 18.1.2.1.2.

#### **Quality of service**

The quality of service of TC request primitives is set by the SACF to the following value:

- sequencing requested;
- return option, this parameter is set by SACF in an implementation-dependent manner.

#### **18.1.1.4 Component handling**

##### **18.1.1.4.1 Procedures for INAP operations**

This subclause describes the procedures for INAP operations.

##### **Operation invocation**

SACF shall build an operation argument from the parameters received and request the invocation of the associated operation using the TC-INVOKE procedure. If a linked ID parameter is inserted in the primitive, this indicates a child operation and implies that the operation is linked to a parent operation.

##### **Operation invocation receipt**

On receipt of a TC-INVOKE indication primitive, SACF shall:

- if the operation code does not correspond to an operation supported by the application-context, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognised operation);
- if a linked ID is included, perform the following checks: If the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to a permitted linked operation, or if the parent operation invocation is not active, issue a TC-U-REJECT request primitive with the appropriate problem code (linked response unexpected or unexpected linked operation);
- if the type of the argument is not the one defined for the operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter);
- if the operation cannot be invoked because the INAP related dialogue is about to be released, request the transfer of the reject component using the TC-U-REJECT request primitive with the problem code (Initiating Release);
- if sufficient INAP related resources are not available to perform the requested operation, request the transfer of a reject component using the TC-U-REJECT request primitive with the problem code (Resource Limitation);
- otherwise, accept the TC-INVOKE indication primitive. If the operation is to be user-confirmed, SACF waits for the corresponding response.

##### **Operation Response**

For user confirmed operations, SACF shall:

- if no error indication is included in the response to a class 1 or 3 operation, construct a result information element from the parameters received and request its transfer using the TC-RESULT-L service;
- if an error indication is included in the response to a class 1 or 2 operation, construct an error parameter from the parameters received and request its transfer using the TC-U-ERROR request primitive.

## Receipt of a response

On receipt of a TC-RESULT-NL indication, SACF shall:

- request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter).

On receipt of a TC-RESULT-L indication, SACF shall:

- if the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter);
- otherwise, accept the TC-RESULT-L indication primitive.

On receipt of a TC-U-ERROR indication, SACF shall:

- if the error code is not defined for the SACF or is not one associated with the operation referred to by the invoke identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognised error or unexpected error);
- if the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter);
- otherwise, accept the TC-U-ERROR indication primitive.

On receipt of a TC-U-REJECT indication primitive which affects a pending operation, SACF shall:

- accept the TC-U-REJECT indication primitive.

On receipt of a TC-L-REJECT indicating "return result problem, return error unexpected", SACF shall inform the application process.

On receipt of a TC-L-REJECT indicating "return error problem, return error unexpected", SACF shall inform the application process.

This event occurs when the local TC detects a protocol error in an incoming component which affects an operation.

When the problem code indicates a general problem, it is considered that the event cannot be related to an active operation even if the invoke Id is provided by TC. This is because it is unclear whether the invoke Id refers to a local or remote invocation. The behaviour of SACF in such a case is described below under the heading "Other events".

On receipt of a TC-L-CANCEL indication, the SACF shall:

- if the associated operation is a class 1 operation, inform the application process;
- if the associated operation is a class 2 operation and no linked operations are defined for this operation, ignore the primitive;
- if the associated operation is a class 2 operation and has linked operations but none of them has been invoked, inform the application process;
- if the associated operation is a class 2 operation and a linked operation invocation has already been received in response to this operation, ignore the primitive;
- if the associated operation is a class 3 operation, inform the application process;
- if the associated operation is a class 4 operation, ignore the primitive.

## Other events

This subclause describes the behaviour of SACF on receipt of a component handling indication primitive which cannot be related to any operation or which does not affect a pending one.

On receipt of a TC-U-REJECT indication primitive which does not affect an active operation (i.e. indicating a return result or return error problem), it is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in 18.1.2.1.2. This is also applicable for invoke problems related to a class 4 linked operation.

On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which does not affect an active operation, it is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in 18.1.2.1.2.

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active operation, it is up to the application process to continue, or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

On receipt of a TC-NOTICE indication primitive, which informs the SACF that a message cannot be delivered by the Network Layer, it is for the application process to decide whether to terminate the dialogue or retry.

This primitive can only occur if the Return Option has been set (see 18.1.1.3.6).

### 18.1.1.4.2 Mapping to TC component primitives

The mapping of parameters onto the TC Component services is as follows:

The TC-U-CANCEL service is not used.

The TC-RESULT-NL service is not used.

The use of parameters of the TC-INVOKE service is as defined in 18.1.1.4.3 with the following qualifications:

- The Operation parameter of the TC-INVOKE service shall contain the *operation.&operationCode* value of the INAP operation to be invoked. The operation must be one of the valid operations supported by the negotiated Application Context for the TC dialogue and must be invokable by the local AE.
- The Parameters parameter of the TC-INVOKE service shall contain a value of the *operation.&ArgumentType* value for the operation being invoked, as specified by the Operation parameter.

The use of parameters of the TC-RESULT-L service is as defined in 18.1.1.4.3 with the following qualifications:

- The Invoke Id parameter of the TC-RESULT-L service shall be set to the value of the Invoke Id parameter of the TC-INVOKE service from the remote AE to which a result is being sent.
- The Operation parameter of the TC-RESULT-L service shall be set to the value of the Operation parameter of the TC-INVOKE service from the remote AE which contains the same Invoke Id Parameter value.
- The Parameters parameter of the TC-RESULT-L service shall contain the *operation.&ResultType* value for the operation result, as specified by the Operation parameter.

The use of parameters of the TC-U-ERROR service is as defined in 18.1.1.4.3 with the following qualifications:

- The Invoke Id parameter of the TC-U-ERROR service shall be set to the value of the Invoke Id parameter of the TC-INVOKE service from the remote AE to which an error is being sent.
- The Error parameter of the TC-U-ERROR service shall be set to the value of the *error.&errorCode* of the error to be sent. It must be one of the errors which is expected for the invoked operation as defined in the *operation.&Errors* specification.
- The Parameters parameter of the TC-U-ERROR service shall be set to the value of the *error.&ParameterType* of the error to be sent, as identified by the Error parameter.

The use of parameters of the TC-U-REJECT service is as defined in 18.1.1.4.3 with the following qualifications:

- The Invoke Id parameter of the TC-U-REJECT service shall be set to the Invoke Id Parameter of the TC component service from the remote AE which is being rejected.

The use of parameters of the TC-L-CANCEL service is as defined in 18.1.1.4.3.

#### **18.1.1.4.3 Default mapping to TC component parameters**

##### **Invoke Id**

This parameter is set by the sending application process. It represents the unique identity of an instance of an operation which is invoked by an AE within a specific TC dialogue. The TC dialogue is identified by the Dialogue Id parameter.

##### **Linked Id**

This parameter is set by the sending application process. It represents the Invoke Id of an operation which was received from the remote AE for a specific TC dialogue to which the operation being invoked by the local AE is to be linked. This parameter is only present if the original operation invoked by the remote AE is defined as having linked operations. The type of local operation invoked must be the same type as one of the operations defined as being linked.

##### **Dialogue Id**

The value of this parameter is associated with the INAP invocation in an implementation-dependent manner. It represents the identity of the established TC dialogue which will carry the component services between the local AE and the remote AE.

##### **Class**

The value of this parameter is set according to the type of the operation to be invoked according to the operation definitions in clauses 5 through 10.

##### **Time-out**

The value of this parameter is set according to the type of operation invoked.

##### **Last component**

This parameter is used as described in Recommendation Q.771.

##### **Problem code**

This parameter is used as described in 18.1.1.4.1.

##### **Abort reason**

This parameter is used by SACF, and attributes and coding are specified by network operator.

## 18.1.2 SSF-SCF interface

### 18.1.2.1 Normal procedures

#### 18.1.2.1.1 SSF-to-SCF messages

This subclause defines the normal procedures for TC messages from the SSF to the SCF.

#### SSF-FSM related messages

A dialogue shall be established when the SSF-FSM moves from the state **Idle** to the state **Waiting for Instructions**. The relevant INAP operation, which can be the InitialDP operation or one of DP-specific operations for TDP-R, shall be transmitted in the same message.

No dialogue shall be established when the SSF-FSM moves from the state **Idle** and back to the state **Idle** on the detection of TDP-N. The relevant INAP operation, which can be the InitialDP operation or one of DP-specific operations for TDP-N, shall be sent with a TC-BEGIN request primitive and the dialogue is locally ended by means of TC-END request primitive with prearranged end.

For all other operations sent from the SSF-FSM, the dialogue shall be maintained except for the following cases.

When the SSF-FSM makes a non-error case state transition to the state **Idle** and there is one or more pending operation and TCAP dialogue is established, TCAP dialogue can be terminated by TC-END primitive with component(s). When the SSF sends the last EventReportBCSM, ApplyChargingReport or CallInformationReport the dialogue may be ended from the SSF by a TC-END request primitive with basic end.

In the case that there is no pending operation and TCAP dialogue is established, TCAP dialogue can be terminated by TC-END primitive with zero component or prearranged end. When the SSF-FSM makes a non-error case state transition to the state **Idle** and there is no operation to be sent, the dialogue is ended by means of a TC-END request primitive (basic) with zero components, or the dialogue is locally ended by means of a TC-END request primitive with prearranged end.

The SSF can end a dialogue with a TC-END request primitive with zero component or prearranged end depending on that TCAP dialogue is established or not, in the case call release is initiated by any other entity than the SCF and the SSF has no pending call information requests (or pending requests which should be treated in the same way, see Note 1, in 11.5.2) nor any armed EDP to notify the SCF of the call release (for alternative way, see 18.1.2.2).

When the SSF has sent the last EventReportBCSM, ApplyChargingReport or CallInformationReport the dialogue may be ended from the SCF by a TC-END request primitive with basic end.

#### Assisting/Hand-off SSF-FSM related messages

A dialogue shall be established when the Assisting/Hand-off SSF-FSM moves from the state **Idle** to the state **Waiting For Instructions**. The AssistRequestInstructions operation shall be transmitted with a TC-BEGIN request primitive.

For all other operations sent from the Assisting/Hand-off SSF-FSM, the dialogue shall be maintained except for the following cases.

When the SSF-FSM makes a non-error case state transition to the state **Idle** and there is one or more pending operation and TCAP dialogue is established, TCAP dialogue can be terminated by TC-END primitive with component(s). When the SSF sends the last ApplyChargingReport, the dialogue may be ended from the SSF by a TC-END request primitive with basic end.



In the case that there is no pending operation and TCAP dialogue is established, TCAP dialogue can be terminated by TC-END primitive with zero component or prearranged end. When the SSF-FSM makes a non-error case state transition to the state **Idle** and there is no operation to be sent, the dialogue is ended by means of a TC-END request primitive (basic) with zero components, or the dialogue is locally ended by means of a TC-END request primitive with prearranged end.

When the SSF has sent the last ApplyChargingReport, the dialogue may be ended from the SCF by a TC-END request primitive with basic end.

### **SSME-FSM related messages**

The following procedures shall be followed:

- The dialogue shall be maintained when the ActivityTest Return Result is sent.
- No dialogue shall be established when the ServiceFilteringResponse operation is sent. The operation is sent with a TC-BEGIN request primitive and the dialogue is ended by means of a TC-END request primitive with prearranged end.
- A dialogue shall no longer be maintained when the Return Result of the ActivateServiceFiltering operation is sent. The dialogue is ended by means of a TC-END request primitive with basic end, the Return Result is transmitted with the same request.
- The dialogue is locally terminated by means of a TC-END request primitive with prearranged end, upon reception of a TC-BEGIN indication primitive with a CallGap operation.
- The dialogue shall be maintained when the RequestCurrentStatusReport, RequestEveryStatusChangeReport or RequestFirstStatusMatchReport operation is received inside the call context.
- The dialogue shall be maintained on sending the following operations inside the call context if the operations are not the final one:
  - Return Result of the RequestCurrentStatusReport operation;
  - Return Result of the RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation; and
  - StatusReport operation in reply to RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation.
- The dialogue shall no longer be maintained on sending the following operations inside the call context if the operations are the final one:
  - Return Result of the RequestCurrentStatusReport operation;
  - Return Result of the RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation; and
  - StatusReport operation in reply to RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation.

The dialogue is ended from the SSF by means of TC-END request primitive with basic end, one of the above operations is transmitted with the same request. If monitor duration expires for RequestFirstStatusMatchReport or RequestEveryStatusChangeReport which has been received inside the call context and there is no need to maintain dialogue, the dialogue is ended from the SSF by means of a TC-END request primitive (basic) with zero component.
- The dialogue shall be established when the RequestCurrentStatusReport, RequestEveryStatusChangeReport or RequestFirstStatusMatchReport operation is received outside the call context.

- The dialogue shall no longer be maintained when the Return Result of the RequestCurrentStatusReport operation is sent outside the call context. The dialogue is ended from the SSF by means of a TC-END request primitive with basic end, the Return Result is transmitted with the same request.
- The dialogue shall be maintained when the Return Result of the RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation is sent outside the call context.
- The dialogue shall no longer be maintained when the StatusReport operation is sent in reply to the RequestFirstStatusMatchReport operation outside the call context. The dialogue is ended from the SSF by means of a TC-END request primitive with basic end, the StatusReport operation is transmitted with the same request.
- The dialogue shall be maintained when the StatusReport operation is sent in reply to the RequestEveryStatusChangeReport operation outside the call context.
- The dialogue shall no longer be maintained when the monitor duration expires for RequestFirstStatusMatchReport or RequestEveryStatusChangeReport which has been received outside the call context. The dialogue is ended from the SSF by means of a TC-END request primitive (basic) with zero component.

#### 18.1.2.1.2 SCF-to-SSF messages

This subclause defines the normal procedures for TC messages from the SCF to the SSF.

##### SCSM-FSM related messages

No dialogue shall be established when the SCSM-FSM moves from state **Idle** to state **Idle** upon receipt of InitialDP operation or one of DP-specific operations for TDP-N. The operation is received with a TC-BEGIN indication primitive and the dialogue is locally terminated by means of a TC-END request primitive with prearranged end.

A dialogue shall be established when the SCSM-FSM moves from state **Idle** to state **Preparing SSF Instructions** upon the receipt of InitialDP operation for TDP-R, one of DP-specific operations for TDP-R or AssistRequestInstructions operation.

A dialogue shall be established when the SCSM-FSM sends an InitiateCallAttempt or a CreateCSA from the **Idle** state.

For subsequent operations sent from the SCSM-FSM, the dialogue shall be maintained except for the following cases, i.e. all other operations are sent after a dialogue was established from the SSF (the SCF has previously received a TC-BEGIN indication primitive with an InitialDP operation, one of DP-specific operations or an AssistRequestInstructions operation).

The dialogue shall no longer be maintained when the prearranged end condition is met in the SCF. When the SCF does not expect any messages other than possibly REJECT or ERROR messages for the operations sent and when the last associated operation timer expires, the dialogue is locally ended by means of a TC-END request primitive with prearranged end.

Alternatively, the sending of operations, leading to the termination of the relationship, by means of a TC-END request primitive (basic end) is possible.

## SCME-FSM related messages

The operations sent from the SCME-FSM shall be issued according to the following procedures:

- The dialogue shall be maintained when the ActivityTest operation is sent.
  - A dialogue shall not be established when a CallGap operation is sent without using a SCSM associated dialogue. The operation is sent using a TC-BEGIN request primitive and the dialogue is terminated with a prearranged end.
  - For sending one or more CallGap operations, the SCME FSM may use an existing SCSM FSM associated dialogue which was initiated by a SSF-FSM (i.e. established for the transmission of the InitialDP operation or one of DP-specific operations). The dialogue shall be maintained and the CallGap operation(s) shall be sent with the first response of the SCSM FSM to the InitialDP operation or one of DP-specific operations.
  - A dialogue shall be established when an ActivateServiceFiltering operation is sent. The operation shall be transmitted with a TC-BEGIN request primitive.
  - The dialogue is locally terminated upon reception of a ServiceFilteringResponse operation using a TC-END request primitive with prearranged end.
  - The dialogue shall be maintained when the RequestCurrentStatusReport, RequestEveryStatusChangeReport or RequestFirstStatusMatchReport operation is sent inside the call context.
  - The dialogue shall be maintained on receiving the following operations inside the call context if the operations are not final one:
    - Return Result of the RequestCurrentStatusReport operation;
    - Return Result of the RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation; and
    - StatusReport operation in reply to RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation.
  - The dialogue shall no longer be maintained on receiving the following operations inside the call context if the operations are final one:
    - Return Result of the RequestCurrentStatusReport operation;
    - RequestEveryStatusChangeReport operation; and
    - StatusReport operation in reply to RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation.
- The dialogue is ended from the SSF by means of TC-END indication primitive with basic end, one of above operations is transmitted with the same request. If monitor duration expires for RequestFirstStatusMatchReport or RequestEveryStatusChangeReport which has been received inside the call context and there is no need to maintain dialogue, the dialogue is ended from the SSF by means of a TC-END indication primitive (basic) with zero component.
- The dialogue shall be established when the RequestCurrentStatusReport, RequestEveryStatusChangeReport or RequestFirstStatusMatchReport operation is sent outside the call context.
  - The dialogue shall no longer be maintained when the Return Result of the RequestCurrentStatusReport operation is received outside the call context. The dialogue is ended from the SSF by means of a TC-END indication primitive with basic end, the Return Result is transmitted with the same indication.

- The dialogue shall be maintained when the Return Result of the RequestFirstStatusMatchReport or RequestEveryStatusChangeReport operation is received outside the call context.
- The dialogue shall no longer be maintained when the StatusReport operation is received in reply to the RequestFirstStatusMatchReport operation outside the call context. The dialogue is ended from the SSF by means of a TC-END indication primitive with basic end, the StatusReport operation is received with the same indication.
- The dialogue shall be maintained when the StatusReport operation is received in reply to the RequestEveryStatusChangeReport operation outside the call context.
- The dialogue shall no longer be maintained when the monitor duration expires for RequestFirstStatusMatchReport or RequestEveryStatusChangeReport outside the call context. The dialogue is ended from the SSF by means of a TC-END indication primitive (basic) with zero component.

### **SCF-SSF – Use of dialogue handling services**

Dialogue handling services are used to trigger the sending of the APDUs associated with the operations involved in the INAP packages.

Component grouping is performed under the control of the application-process through an appropriate usage of the TC-BEGIN and TC-CONTINUE service.

The TC-END service is solely used to support the dialogue closing procedure (i.e. it is never used to trigger the sending of components).

On receipt of an empty TC-CONTINUE.req primitive, the FE should ignore the primitive.

On receipt of a TC-END.req with an INAP request, the FE should not perform the request and consider the requested TC-END service as a dialogue closing procedure. The dialogue is then terminated.

It is an application-process responsibility to provide in the TC-BEGIN.req primitive a destination address which can be used by the underlying SCCP to route the message to the proper FE if this FE is addressed through the SS7 network.

The prearranged end can be used.

#### **18.1.2.2 Abnormal procedures**

The following procedures also apply to the SCF-SRF and SCF-CUSF interfaces.

#### **SCF-to-SSF/SRF/CUSF messages**

Considering that SSF, SRF, and CUSF do not have the logic to recover from error cases detected on the SCF-SSF/SRF/CUSF interface, the following shall apply:

- Operation errors and rejection of TCAP components shall be transmitted to the SSF and, respectively, the SRF, and the CUSF with a TC-END request primitive, basic end.

If, in violation of the above procedure, an ERROR or REJECT component is received with a TC-CONTINUE indication primitive, the SSF and, respectively, the SRF and the CUSF shall abort the dialogue with a TC-U-ABORT request primitive.

In the case of the SSF relay, it is for further study how to map messages to ROSE capability of bearer signalling system between the SSF and the SRF, and what services are assumed from ROSE.

## **SSF/SRF/CUSF-to-SCF messages**

Operation errors and rejection of TCAP components shall be transmitted to the SCF according to the following rules:

- The dialogue shall be maintained when the preceding message, which contained the erroneous component, indicated that the dialogue shall be maintained, i.e. the error or reject shall be transmitted with a TC-CONTINUE request primitive if the erroneous component was received with a TC-CONTINUE indication primitive.  
On receipt of an ERROR or REJECT component, the SCF decides on further processing. It may either continue, explicitly end or abort the dialogue.
- In all other situations the dialogue shall no longer be maintained, i.e. the error or reject shall be transmitted with a TC-END request primitive, basic end, if the erroneous component was received with a TC-BEGIN indication primitive.
- on expiration of application timer  $T_{SSF}$  or  $T_{SRF}$  or  $T_{CUSF}$ , dialogue shall be terminated by means of by TC-U-ABORT primitive with an Abort reason, regardless of whether TCAP dialogue is established or not.

If the error processing in the SSF/SRF/CUSF leads to the case where the SSF/SRF/CUSF is not able to process further SCF operations while the dialogue is to be maintained, the SSF/SRF/CUSF aborts the dialogue with a TC-END request primitive with basic end or a TC-U-ABORT request primitive, depending on whether any pending ERROR or REJECT component is to be sent or not.

The SSF can end a dialogue with a TC-U-ABORT request primitive in case call release is initiated by any other entity then the SCF and the SSF has no pending call information requests (or pending requests which should be treated in the same way, i.e. ApplyCharging nor any armed EDP to notify the SCF of the call release (for alternative way, see 18.1.2.1.1).

In the case of the SSF relay, it is for further study how to map messages to ROSE capability of bearer signalling system between the SSF and the SRF, and what services are assumed from ROSE.

The CUSF can end a dialogue with a TC-U-ABORT request primitive in case the association release between the user and the network is initiated by any other entity.

## **SCF-SSF – Use of dialogue handling services**

On receipt of a TC-U-REJECT.ind in the FE, this primitive should be ignored. It is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in 18.1.2.2. This is also applicable for invoke problems related to a class 4 linked operation.

A TC-U-REJECT.req should be sent followed by a TC-CONTINUE.req.

On receipt of a TC-R-REJECT.ind in the FE, this primitive should be ignored. It is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in 18.1.2.2. This is also applicable for invoke problems related to a class 4 linked operation. The dialogue should be released with a TC-U-ABORT.req.

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active operation, it is up to the application process to continue or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

On receipt of a TC-NOTICE indication the SACF is informed that a message cannot be delivered by the Network Layer. It occurs if the Return Option has been set (see 18.1.2.2). It is for the application process to decide whether to terminate the dialogue or retry.

The application-process is the sole user of the TC-P-ABORT service and TC-NOTICE service.

The receipt of a TC-U-ABORT.ind or TC-P-ABORT.ind on a dialogue terminates all request processing.

### 18.1.2.3 Dialogue handling

#### 18.1.2.3.1 Dialogue establishment

#### 18.1.2.3.2 Dialogue continuation

#### 18.1.2.3.3 Dialogue termination

#### 18.1.2.3.4 User Abort

#### 18.1.2.3.5 Provider Abort

#### 18.1.2.3.6 Mapping to TC dialogue primitives

The SSF-SCF IN services can be mapped onto TC services. This subclause defines the mapping of the SSF-SCF IN services onto the services of the TC dialogue handling services defined in Recommendation Q.771.

- a) The TC-BEGIN service is used to invoke the operations of the **scf-ssfConnectionPackage** and **ssf-scfConnectionPackage**.
- b) The TC-CONTINUE service is used to report the success of the operations invoked in a TC-BEGIN service and to invoke or respond to any other operations.
- c) The TC-U-ABORT service is used to report the failure of operations of the **scf-ssfConnectionPackage** and **ssf-scfConnectionPackage**.

The mapping of the parameters onto the TC-BEGIN primitive is defined in 18.1.1.3.6 with the following qualifications:

- The Application Context Name parameter shall take the value of the application-context-name field of the **cs2ssf-scfGenericAC**, **cs2ssf-scfDPSpecificAC**, **cs2ssf-scfAssistHandoffAC** or **cs2ssf-scfServiceManagementAC** object if the initiating AE is a SSF or the **cs2scf-ssfGenericAC**, **cs2scf-ssfDPSpecificAC**, **cs2scf-ssfTrafficManagementAC**, **cs2scf-ssfServiceManagementAC** or **cs2scf-ssfStatusReportingAC** object if the originating AE is a SCF.

The mapping of the parameters onto the TC-CONTINUE primitive is defined in 18.1.1.3.6.

The mapping of the parameters onto the TC-U-ABORT primitive is defined in 18.1.1.3.6 with the following qualifications:

- The Application-Context-Name parameter shall be used as specified in Recommendation Q.771. When the responding AE refuses a dialogue because the application-context-name it receives is not supported, this parameter shall have the value of the application-context-name field of the **cs2ssf-scfGenericAC**, **cs2ssf-scfDPSpecificAC**, **cs2ssf-scfAssistHandoffAC**, **cs2ssf-scfServiceManagementAC** or **cs2scf-ssfGenericAC** object if the responding AE is a SCF or the **cs2scf-ssfDPSpecificAC**, **cs2scf-ssfTrafficManagementAC**, **cs2scf-ssfServiceManagementAC** or **cs2scf-ssfStatusReportingAC** object if the responding AE is a SSF.

The use of the parameters of the TC-END service is defined in 18.1.1.3.6.

#### **18.1.2.4 Component handling**

##### **18.1.2.4.1 Procedures for INAP operations**

The INAP ASEs are users of the TC component handling services except for the TC-L-REJECT and TC-L-CANCEL services which are used by the application-process. Receipt of a TC-L-REJECT.ind leads the application-process to abandon the dialogue (i.e. it issues a TC-U-ABORT.request primitive).

The TC-U-CANCEL service is never used.

##### **18.1.2.4.2 Mapping to TC component parameters**

The SSF-SCF IN ASE services are mapped onto the TC component handling services. The mapping of operations and errors onto TC services is defined in 18.1.1.4.2 with the following qualification:

The time-out parameter of the TC-INVOKE.req primitives is set according to Table 5-1.

#### **18.1.3 SCF-SRF interface**

##### **18.1.3.1 Normal procedures**

###### **SCF-to/from-SRF messages**

A dialogue is established when the SRF sends an AssistRequestInstructions operation to the SCF. For all other operations sent to/from the SRF, the dialogue shall be maintained.

In the case that there is no pending operation and TCAP dialogue is established, TCAP dialogue can be terminated by TC-END primitive with zero component. When the SCSM makes a non-error case state transition to end-user interaction and there is no operation to be sent, the dialogue is ended by means of a TC-END request primitive (basic) with zero components.

The dialogue shall no longer be maintained when sending the SpecializedResourceReport operation for PlayAnnouncement with disconnection from the SRF set to TRUE or Return Result of the PromptAndCollectUserInformation with disconnection from the SRF set to TRUE. The dialogue is ended by means of a TC-END request primitive with basic end, and one of the above operations is transmitted with the same request.

Regardless of whether pending operation exists or not, when the SRSF-FSM is informed of the disconnection of bearer connection (in the case of SCF initiated disconnection or call abandon from call party) and dialogue is established, the dialogue is ended by means of a TC-END request primitive (basic) with zero components or TC-END request primitive (prearranged end).

The dialogue shall no longer be maintained when the prearranged end condition is met in the SRF. When the SRSF-FSM is informed the disconnection of bearer connection and TCAP dialogue is not established, TCAP dialogue is locally terminated by TC-END primitive with prearranged end.

When the SCF does not expect any messages other than possibly REJECT or ERROR messages for the operations sent and when the last associated operation timer expires, the dialogue is locally ended by means of a TC-END request primitive with prearranged end. Alternatively, the sending of operations, leading to the termination of the relationship, by means of a TC-END request primitive (basic end) is possible.

In the relay case, the SRF-SCF relationship uses the SSF-SCF TCAP dialogue. This is possible, because begin and end of the SRF-SCF relationship are embedded in the SSF-SCF relationship. SRF-SCF information shall be exchanged with TC-CONTINUE request primitives.

In the case of the SSF relay, it is for further study how to map messages to ROSE capability of bearer signalling system between the SSF and the SRF, and what services are assumed from ROSE.

### 18.1.3.2 Abnormal procedures

### 18.1.3.3 Dialogue handling

#### 18.1.3.3.1 Dialogue establishment

#### 18.1.3.3.2 Dialogue continuation

#### 18.1.3.3.3 Dialogue termination

#### 18.1.3.3.4 User Abort

#### 18.1.3.3.5 Provider Abort

#### 18.1.3.3.6 Mapping to TC dialogue primitives

The SCF-SRF IN services can be mapped onto TC services. This subclause defines the mapping of the SCF-SRF IN services onto the services of the TC dialogue handling services defined in Recommendation Q.771.

- a) The TC-BEGIN service is used to invoke the operations of the **srf-scfConnectionPackage**.
- b) The TC-CONTINUE service is used to report the success of the operations invoked in a TC-BEGIN service and to invoke or respond to any other operations.
- c) The TC-U-ABORT service is used to report the failure of operation of the **scf-srfOperationPackage** and **srf-scfOperationPackage**.

The mapping of parameters onto the TC Dialogue services is as defined in 18.1.1.3.6.

The mapping of the parameters onto the TC-BEGIN primitive is defined in 18.1.1.3.6 with the following qualification:

- The Application Context Name parameter shall take the value of the application-context-name field of the **srf-scf-ac** object.

### 18.1.3.4 Component handling

#### 18.1.3.4.1 Procedures for INAP operations

#### 18.1.3.4.2 Mapping to TC component parameters

The mapping of parameters for the TC component services is defined in 18.1.1.4.2 with the following qualification:

The Time-out Parameter of the TC-INVOKE service is set according to Table 6-1.

### 18.1.4 SCF-CUSF interface

#### 18.1.4.1 Normal procedures

##### 18.1.4.1.1 CUSF-to-SCF messages

##### CUSF-FSM related messages

A dialogue shall be established when the CUSF-FSM moves from the state **Idle** to the state **Waiting for Instructions**. The relevant INAP operation, which is one of operations for TDP-R, shall be transmitted in the same message.



No dialogue shall be established when the CUSF-FSM moves from the state **Idle** and back to the state **Idle** on the detection of TDP-N. The relevant INAP operation, which is one of operations for TDP-N, shall be sent with a TC-BEGIN request primitive and the dialogue is locally ended by means of TC-END request primitive with prearranged end.

For all other operations sent from the CUSF-FSM, the dialogue shall be maintained except for the following cases.

When the CUSF sends the last event report operation (currently only ComponentReceived is applicable), the dialogue may be ended from the CUSF by a TC-END request primitive with basic end.

In the case that there is no pending operation and TCAP dialogue is established, TCAP dialogue can be terminated by TC-END primitive with zero component or prearranged end. When the CUSF-FSM makes a non-error case state transition to the state **Idle** and there is no operation to be sent, the dialogue is ended by means of a TC-END request primitive (basic) with zero components, or the dialogue is locally ended by means of a TC-END request primitive with prearranged end. The CUSF can end a dialogue with a TC-END request primitive with zero component or prearranged end depending on whether TCAP dialogue is established or not, in the case association release between the user and the network is initiated by any other entity.

When the CUSF has sent the last event report operation (currently only ComponentReceived is applicable), the dialogue may be ended from the SCF by a TC-END request primitive with basic end.

#### **CUSME-FSM related messages**

The dialogue shall be maintained when the ActivityTest Return Result is sent.

##### **18.1.4.1.2 SCF-to-CUSF messages**

#### **SCSM-FSM related messages**

No dialogue shall be established when the SCSM-FSM moves from state **Idle** to state **Idle** upon receipt of one of DP-specific operations for TDP-N. The operation is received with a TC-BEGIN indication primitive and the dialogue is locally terminated by means of a TC-END request primitive with prearranged end.

A dialogue shall be established when the SCSM-FSM moves from state **Idle** to state **Preparing CUSF Instructions** upon the receipt of one of operations for TDP-R.

For subsequent operations sent from the SCSM-FSM, the dialogue shall be maintained except for the following cases, i.e. all other operations are sent after a dialogue was established from the CUSF (the SCF has previously received a TC-BEGIN indication primitive with one of the operations for TDP-R).

The dialogue shall no longer be maintained when the prearranged end condition is met in the SCF. When the SCF does not expect any messages other than possibly REJECT or ERROR messages for the operations sent and when the last associated operation timer expires, the dialogue is locally ended by means of a TC-END request primitive with prearranged end.

Alternatively, the sending of operations, leading to the termination of the relationship, by means of a TC-END request primitive (basic end) is possible.

#### **SCME-FSM related messages**

The operation(s) sent from the SCME-FSM shall be issued according to the following procedure(s):

- The dialogue shall be maintained when the ActivityTest operation is sent.

#### **18.1.4.2 Abnormal procedures**

#### **18.1.4.3 Dialogue handling**

##### **18.1.4.3.1 Dialogue establishment**

##### **18.1.4.3.2 Dialogue continuation**

##### **18.1.4.3.3 Dialogue termination**

##### **18.1.4.3.4 User Abort**

##### **18.1.4.3.5 Provider Abort**

##### **18.1.4.3.6 Mapping to TC dialogue primitives**

The CUSF-SCF IN services can be mapped onto TC services. This subclause defines the mapping of the CUSF-SCF IN services onto the services of the TC dialogue handling services defined in Recommendation Q.771.

- a) The TC-BEGIN service is used to invoke the operations of the **cusf-scfConnectionPackage**.
- b) The TC-CONTINUE service is used to report the success of the operations invoked in a TC-BEGIN service and to invoke or respond to any other operations.
- c) The TC-U-ABORT service is used to report the failure of operation of the **cusf-scfOperationPackage** and **scf-cusfOperationPackage**.

The mapping of parameters onto the TC Dialogue services is as defined in 18.1.1.3.6.

The mapping of the parameters onto the TC-BEGIN primitive is defined in 18.1.1.3.6 with the following qualification:

- The Application Context Name parameter shall take the value of the application-context-name field of the **cusf-scf-ac** object if the originating AE is a CUSF or the **scf-cusf-ac** object if the originating AE is a SCF.

#### **18.1.4.4 Component handling**

##### **18.1.4.4.1 Procedures for INAP operations**

##### **18.1.4.4.2 Mapping to TC component parameters**

The mapping of parameters for the TC component services is defined in 18.1.1.4.2 with the following qualification:

The Time-out Parameter of the TC-INVOKE service is set according to Table 10-1.

#### **18.1.5 SCF-SCF interface**

##### **18.1.5.1 Normal procedures**

Dialogue handling services are used to trigger the sending of the APDUs associated with the operations involved in the INAP packages.

Component grouping is performed under the control of the application-process through an appropriate usage of the TC-BEGIN and TC-CONTINUE service.

The TC-END service is solely used to support the dialogue closing procedure (i.e. it is never used to trigger the sending of components).

On receipt of an empty TC-CONTINUE.ind primitive, the application process should ignore the primitive.

The prearranged end can be used.

It is an application-process responsibility to provide in the TC-BEGIN.req primitive a destination address which can be used by the underlying SCCP to route the message to the proper FE if this FE is addressed through the SS7 network.

#### **18.1.5.2 Abnormal procedures**

On receipt of a TC-U-REJECT.ind in the FE, this primitive should be ignored. It is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in 18.1.1.4.1. This is also applicable for invoke problems related to a class 4 linked operation.

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active operation, it is up to the application process to continue or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

On receipt of a TC-NOTICE indication the SACF is informed that a message cannot be delivered by the Network Layer. It occurs if the Return Option has been set (see 18.1.1.3.7). It is for the application process to decide whether to terminate the dialogue or retry.

The receipt of a TC-U-ABORT.Ind or TC-P-ABORT.Ind on a dialogue terminates all request processing.

#### **18.1.5.3 Dialogue handling**

##### **18.1.5.3.1 Dialogue establishment**

##### **18.1.5.3.2 Dialogue continuation**

##### **18.1.5.3.3 Dialogue termination**

##### **18.1.5.3.4 User Abort**

##### **18.1.5.3.5 Provider Abort**

If the Security Exchange Service Element (SESE) is included in the application context, and an SE-P-ABORT is required, then the SE-P-ABORT is mapped to the TC-U-ABORT primitive.

##### **18.1.5.3.6 Mapping to TC dialogue primitives**

The SCF-SCF IN services can be mapped onto TC services. This subclause defines the mapping of the SCF-SCF IN services onto the services of the TC dialogue handling services defined in Recommendation Q.771.

- a) The TC-BEGIN service is used to invoke the operations of the **scf-scfConnectionPackage** and the **dsdpConnectionPackage** and **SETransfer** operations.
- b) The TC-CONTINUE service is used to report the success of the operations invoked in a TC-BEGIN service and to invoke or respond to any other operations. If the SESE is included in the application context, the TC-CONTINUE service is used for the second and third exchanges.
- c) The TC-U-ABORT service is used to report the failure of operation of the **scf-scfConnectionPackage** and **dsdpConnectionPackage** and **SETransfer** operations.

The mapping of the parameters onto the TC-service is as follows:

The use of the parameters of the TC-BEGIN service is as defined in 18.1.1.3.6 with the following qualifications:

- The Application Context Name parameter of the TC-BEGIN service shall take the value of the `application-context-name` field of the **scf-scfOperationsAC** or **distributedSCFSystemAC** object.
- The use of the User Information parameter of the TC-BEGIN service shall contain an EXTERNAL ASN.1 Type with the direct-reference field, if present, set to **id-as-scfBindingAS** and the value to be encoded shall be of type **SCF-SCFBinding-PDUs.bind.bind-invoke (SCFBindRequestArgument)**. If the SESE is included in the application context, the User Information parameter of the TC-BEGIN service shall contain a value of type **seItem**.

The use of the parameters of the TC-CONTINUE service is as defined in 18.1.1.3.6 with the following qualifications:

- The use of the User Information parameter of the first TC-CONTINUE service shall contain an EXTERNAL ASN.1 Type with the direct-reference field, if present, set to **id-as-scfBindingAS** and the value to be encoded shall be of type **SCF-SCFBinding-PDUs.bind.bind-result (SCFBindResultArgument)**. If the SESE is included in the application context, the User Information parameter of the TC-CONTINUE service shall contain a value of type **seItem**.

In subsequent TC-CONTINUE services, the use of the User Information field is network-operator specific.

The use of the parameters of the TC-U-ABORT service is as defined in 18.1.1.3.6 with the following qualifications:

- The Application-Context-Name parameter of the TC-U-ABORT service shall be used as specified in Recommendation Q.771. When the SCF refuses a dialogue because the application-context-name it receives is not supported, the value of the application-context-name it returns is selected as follows:
  - if the value has the same root as the **scf-scfOperationsAC**, the **scf-scfOperationsAC** is used;
  - if the value has the same root as the **distributedSCFSystemAC**, the **distributedSCFSystemAC** is used;
  - otherwise the received value is returned.

The use of the parameters of the TC-END service is as defined in 18.1.1.3.6.

#### 18.1.5.4 Component handling

##### 18.1.5.4.1 Procedures for INAP operations

The INAP ASEs are users of the TC component handling services except for the TC-L-REJECT and TC-L-CANCEL services which are used by the application-process. Receipt of a TC-L-REJECT.ind leads the application-process to abandon the dialogue (i.e. it issues a TC-U-ABORT.request primitive).

The TC-U-CANCEL service is never used.

#### **18.1.5.4.2 Mapping to TC component parameters**

The SCF-SCF IN ASE services are mapped onto the TC component handling services. The mapping of operations and errors onto TC services is defined in 18.1.1.4.2 with the following qualification:

The time-out parameter of the TC-INVOKE.req primitives is set according to Table 9-1. The SCFBind timer ( $T_{bi}$ ) does not map to a TC timer.

#### **18.1.6 SCF-SDF interface**

All the services provided by the Directory ASEs are contained in a single AE. The Component Handler (CHA) of the Transaction Capabilities (TC) supports the request/reply paradigm of the operation. The Directory ASEs provide the mapping function of the abstract-syntax notation of the directory operation packages onto the services provided by TC. The Dialogue Handler (DHA) of the TC supports the establishment and release of an application-association called "dialogue" between a pair of AEs. Dialogues between a DUA and a DSA may be established only by the DUA.

##### **18.1.6.1 Normal procedures**

Dialogue handling services are used to support the DirectoryBind and DirectoryUnbind operations and to trigger the sending of the APDUs associated with the operations involved in the Directory packages.

Component grouping is performed under the control of the application-process through an appropriate usage of the TC-BEGIN and TC-CONTINUE service.

The TC-END service is solely used to support the unbind procedure (i.e. it is never used to trigger the sending of components).

On receipt of an empty TC-CONTINUE.req primitive, the SDF should ignore the primitive.

On receipt of an TC-END.req with a database request, the SDF should not perform the database request and consider the requested TC-END service as an unbind procedure. The dialogue is then terminated.

##### **18.1.6.2 Abnormal procedures**

On receipt of a TC-U-REJECT.ind in the SDF, this primitive should be ignored.

On receipt of a TC-R-REJECT.ind in the SDF, the dialogue should be released with a TC-U-ABORT.req.

If reject situations are detected in the SDF, a TC-U-REJECT.req should be sent followed by a TC-CONTINUE.req.

The prearranged termination procedure is never used.

The application-process is the sole user of the TC-P-ABORT service and TC-NOTICE service.

The receipt of a TC-U-ABORT.ind or TC-P-ABORT.ind on a dialogue terminates all request processing. It is the application-process responsibility to confirm if requested modifications occurred.

It is an application-process responsibility to provide in the TC-BEGIN.req primitive a destination address which can be used by the underlying SCCP to route the message to the proper SDF.

### 18.1.6.3 Dialogue handling

#### 18.1.6.3.1 Dialogue establishment

#### 18.1.6.3.2 Dialogue continuation

#### 18.1.6.3.3 Dialogue termination

#### 18.1.6.3.4 User Abort

If the SESE is included in the application context, and an SE-U-ABORT is required, then the SE-U-ABORT is mapped to the TC-U-ABORT primitive.

#### 18.1.6.3.5 Provider Abort

If the SESE is included in the application context, and an SE-P-ABORT is required, then the SE-P-ABORT is mapped to the TC-U-ABORT primitive.

#### 18.1.6.3.6 Mapping to TC dialogue primitives

This subclause defines the mapping of the DirectoryBind and DirectoryUnbind services onto the services of the TC dialogue handling services defined in Recommendation Q.771.

#### Bind

The DirectoryBind service is mapped onto TC-services as follows:

- a) The TC-BEGIN service is used to invoke the **DirectoryBind** operation and SETransfer operations.
- b) The TC-CONTINUE service is used to report the success of the **DirectoryBind** operation. If the SESE is included in the application context, the TC-CONTINUE service is used for the second and third exchanges.
- c) The TC-U-ABORT service is used to report the failure of the **DirectoryBind** operation and SETransfer operations.

The use of the parameter of these services is qualified in the following subclauses.

The use of parameters of the TC-BEGIN service is defined in 18.1.1.3.6 with the following qualifications:

- The Application Context Name parameter of the TC-BEGIN service shall take the value of the application-context-name field of the **iNdirectoryAccessAC** or **inExtendedDirectoryAccessAC** object.
- The Dialogue ID parameter of the TC-CONTINUE service shall be used as specified in 18.1.1.3.7.  
The Authorized Relationship Id can be mapped onto the TC Dialogue ID parameter.
- The User Information parameter of the TC-BEGIN service shall contain an EXTERNAL ASN.1 Type with the direct-reference field, if present, set to **id-as-indirectoryBindingAS** and the value to be encoded shall be of type **DAPBinding-PDUs.bind.bind-invoke (DirectoryBindArgument)**.

If the SESE is included in the application context, the User Information parameter of the TC-BEGIN service shall contain a value of type **seItem**.

The use of parameters of the TC-CONTINUE service is defined in 18.1.1.3.6 with the following qualifications:

- The Dialogue ID parameter of the TC-CONTINUE service shall be used as specified in 18.1.1.3.7.

The Authorized Relationship Id can be mapped onto the TC Dialogue ID parameter.

- The User Information parameter of the TC-CONTINUE service shall contain an EXTERNAL ASN.1 Type with the direct-reference field, if present, set to **id-as-indirectoryBindingAS** and the value to be encoded shall be of type **DAPBinding-PDUs.bind.bind-result (DirectoryBindResult)**.

If the SESE is included in the application context, the User Information parameter of the TC-CONTINUE service shall contain a value of type **seItem**.

The use of parameters of the TC-U-ABORT service is defined in 18.1.1.3.6 with the following qualifications:

- The Dialogue ID parameter of the TC-CONTINUE service shall be used as specified in 18.1.1.3.7.

The Authorized Relationship Id can be mapped onto the TC Dialogue ID parameter.

- The Application Context Name parameter of the TC-U-ABORT service shall be set as defined in 18.1.1.3.6 unless the TC-U-ABORT is reporting the failure of the directoryBind operation. In this case it is set to the value of the application-context-name field of the **iNdirectoryAccessAC** or **inExtendedDirectoryAccessAC** object.
- The User Information parameter of the TC-U-ABORT service shall be used as specified in 18.1.1.3.7, unless the TC-U-ABORT is reporting the failure of a directoryBind operation. In this case this parameter shall contain an EXTERNAL ASN.1 Type with the direct-reference field, if present, set to **id-as-indirectoryBindingAS** and the value to be encoded shall be of type **DAPBinding-PDUs.bind.bind-error (DirectoryBindError)**.

## Unbind

The **DirectoryUnbind** service is mapped onto the TC-END service.

The use of the parameters of the TC-END service is defined in 18.1.1.3.6 with the following qualifications:

- The Dialogue ID parameter of the TC-CONTINUE service shall be used as specified in 18.1.1.3.7.

The Authorized Relationship Id can be mapped onto the TC Dialogue ID parameter.

## 18.1.6.4 Component handling

### 18.1.6.4.1 Procedures for INAP operations

The Directory ASEs are users of the TC component handling services except for the TC-L-REJECT and TC-L-CANCEL services which are used by the application-process. Receipt of a TC-L-REJECT.ind leads the application-process to abandon the dialogue (i.e. it issues a TC-U-ABORT.request primitive).

The TC-U-CANCEL service is never used.

#### 18.1.6.4.2 Mapping to TC component parameters

The Directory ASE services are mapped onto the TC component handling services. The mapping of operations and errors onto TC services is defined in 18.1.1.4.2 with the following qualification:

The time-out parameter of the TC-INVOKE.req primitives is set according to Table 18-2.

**Table 18-2/Q.1228 – TC timer values  
of DAP operations**

Operation	Time-out
search	Medium
modifyEntry	Medium
addEntry	Medium
removeEntry	Medium
execute	Medium

#### 18.1.7 SDF-SDF interface

##### 18.1.7.1 Normal procedures

##### 18.1.7.2 Abnormal procedures

##### 18.1.7.3 Dialogue handling

##### 18.1.7.3.1 Dialogue establishment

##### 18.1.7.3.2 Dialogue continuation

##### 18.1.7.3.3 Dialogue termination

##### 18.1.7.3.4 User Abort

If the SESE is included in the application context, and an SE-U-ABORT is required, then the SE-U-ABORT is mapped to the TC-U-ABORT primitive.

##### 18.1.7.3.5 Provider Abort

If the SESE is included in the application context, and an SE-P-ABORT is required, then the SE-P-ABORT is mapped to the TC-U-ABORT primitive.

##### 18.1.7.3.6 Mapping to TC dialogue primitives

The DSP and DISP can be mapped onto TC services. This subclause defines the mapping of the **DSABind**, **DSAUnbind**, **DSAShadowBind** and **DSAShadowUnbind** services onto the services of the TC dialogue handling services defined in Recommendation Q.771.

The **DirectoryBind** service is mapped onto TC-services as follows:

- The TC-BEGIN service is used to invoke the **DSAShadowBind** and **DSABind** operations.
- The TC-CONTINUE service is used to report the success of the **DSAShadowBind** and **DSABind** operations.
- The TC-U-ABORT service is used to report the failure of the **DSAShadowBind** and **DSABind** operations.



The mapping of parameters onto the TC Dialogue services is as follows:

The use of parameters of the TC-BEGIN service is defined in 18.1.1.3.6 with the following qualifications:

- The Application Context Name parameter of the TC-BEGIN service shall take the value of the application-context-name field of the **inDirectorySystemAC** or **shadowSupplierInitiatedAC** or **shadowConsumerInitiatedAC** object.
- The User Information parameter of the TC-BEGIN service shall contain a value of type EXTERNAL which depends on the value of the Application Context being used.

If the Application Context being used is **inDirectorySystemAC** then the User Information parameter shall contain an EXTERNAL Type with the direct-reference field, if present, set to **id-as-indirectoryDSABindingAS** and the value to be encoded shall be of type **DSABinding-PDUs.bind.bind-invoke**.

If the Application Context being used is **shadowSupplierInitiatedAC** or **shadowConsumerInitiatedAC** then the User Information parameter shall contain an EXTERNAL Type with the direct-reference field, if present, set to **id-as-indsaShadowBindingAS** and the value to be encoded shall be of type **DISBinding-PDUs.bind.bind-invoke**.

The use of parameters of the TC-CONTINUE service is defined in 18.1.1.3.6 with the following qualifications:

- The Dialogue Id parameter of the TC-CONTINUE service shall be used as specified in Recommendation Q.771. The Authorized Relationship Id can be mapped onto the TCAP Dialogue Id.
- The User Information parameter of the first TC-CONTINUE service shall contain a value of type EXTERNAL which depends on the value of the Application Context being used.

If the Application Context being used is **inDirectorySystemAC** then the User Information parameter shall contain an EXTERNAL Type with the direct-reference field, if present, set to **id-as-indirectoryDSABindingAS** and the value to be encoded shall be of type **DSABinding-PDUs.bind.bind-result**.

If the Application Context being used is **shadowSupplierInitiatedAC** or **shadowConsumerInitiatedAC** then the User Information parameter shall contain an EXTERNAL Type with the direct-reference field, if present, set to **id-as-indsaShadowBindingAS** and the value to be encoded shall be of type **DISPBinding-PDUs.bind.bind-result**.

The use of parameters of the TC-U-ABORT service is defined in 18.1.1.3.6 with the following qualifications:

- The Dialogue Id parameter of the TC-U-ABORT service shall be used as specified in 18.1.1.3.6. The Authorized Relationship Id can be mapped onto the TCAP Dialogue Id.
- The Application-Context-Name parameter of the TC-U-ABORT service shall be used as specified in Recommendation Q.771. When the SDF refuses a dialogue because the application-context-name it receives is not supported, this parameter shall have the value of the application-context-name field of the **inDirectorySystemAC** or **shadowSupplierInitiatedAC** or **shadowConsumerInitiatedAC** object.

- When the TC-U-ABORT is reporting the failure of a bind operation, i.e. the abort reason parameter has the value "dialogue-refused", the User Information parameter of the TC-U-ABORT service shall contain a value of type EXTERNAL which depends on the value of the Application Context being used.

If the Application Context being used is **inDirectorySystemAC** then the User Information parameter shall contain an EXTERNAL Type with the direct-reference field, if present, set to **id-as-indirectoryDSABindingAS** and the value to be encoded shall be of type **DSABinding-PDUs.bind.bind-error**.

If the Application Context being used is **shadowSupplierInitiatedAC** or **shadowConsumerInitiatedAC** then the User Information parameter shall contain an EXTERNAL Type with the direct-reference field, if present, set to **id-as-indsaShadowBindingAS** and the value to be encoded shall be of type **DISPBinding-PDUs.bind.bind-error**.

Otherwise it shall be absent.

The **inDSAUnbind** and **inDSAShadowUnbind** services are mapped onto the TC-END service. The use of the parameters of the TC-END service is defined in 18.1.1.3.6.

The **SETTransfer** service is mapped onto TC-services is the same as for the other services except:

- If the SESE is included in the application context, the User Information parameter of the TC-CONTINUE service shall contain a value of type **seItem**.

#### 18.1.7.4 Component handling

##### 18.1.7.4.1 Procedures for INAP operations

##### 18.1.7.4.2 Mapping to TC component parameters

The Directory ASE services are mapped onto the TC component handling services. The mapping of operations and errors onto TC services is defined in 18.1.1.4.2 with the following qualification:

The time-out parameter of the TC-INVOKE.req primitives is set according to Table 18-3.

**Table 18-3/Q.1228 – TC timer values  
of DSP/DISP operations**

Operation	Time-out
chainedAddEntry	Medium
chainedRemoveEntry	Medium
chainedModifyEntry	Medium
chainedExecute	Medium
chainedSearch	Medium
updateShadow	Medium
coordinateShadowUpdate	Medium
requestShadowUpdate	Medium

## **18.2 Services assumed from SCCP**

This subclause describes the services required from the SCCP that may be used by the IN applications for the IN Application Protocol used between the SSF, SCF and SRF, SDF and CUSF.

The services described are those given in the SCCP post *White Book* Recommendations Q.711-Q.714 (07/96) and Q.715 (SCCP User Guide) should be consulted to identify possible interworking and compatibility issues between the different SCCP versions.

### **18.2.1 Normal procedures**

The SCCP forms the link between the TC and the MTP and provides (in conjunction with the MTP) the network services for the IN applications. The network services provided allow the signalling messages sent by the application to the lower layers to be successfully delivered to the peer application.

### **18.2.2 Service functions from SCCP**

#### **18.2.2.1 SCCP connectionless services**

The following Connectionless services are expected from the SCCP:

- a) Network Addressing to enable signalling connections between SCCP users.
- b) Sequence Control to enable the SCCP users to invoke "sequence guaranteed" or "sequence not guaranteed" options for a given stream of messages to the same destination.
- c) Segmentation/reassembly of large user messages.
- d) Return Option to enable the SCCP users to invoke "discard message on error" or "return message on error" for a given message not able to be delivered by the SCCP to the destination SCCP user, due to routing or segmentation/reassembly failure.
- e) Congestion control.

The primitives used for the above services are given below.

The N-UNITDATA request and N-UNITDATA indication primitives are used to send and receive data. The parameters of these primitives include the Called and Calling Addresses, Sequence Control, Return Option and User Data with the addressing parameters always mandatory.

The N-NOTICE indication primitive is used to return undelivered data if return option is set and a routing/segmentation error occurs.

#### **18.2.2.1.1 INAP addressing**

The INAP addressing elements consist of information contained within the Calling and the Called Addresses which are sent by the application to TC for use by SCCP.

The application expects the SCCP to route messages by either:

- a) the use of the Destination Point Code (DPC), the SubSystem Number (SSN) and MTP SAP (Service Access Point) instance; or
- b) the use of the Global Title (GT) plus optionally the SSN, DPC and MTP SAP.

The application also specifies to SCCP whether to use Route on SSN or Route on GT for both the Called and Calling Addresses.

If INAP requires additional addressing information, it must be carried in the GT portion of the address specification regardless of which form of routing is specified.

Method a) above may be used when the application is aware of the destination point code and the destination SSN located at that point code to which the message is to be delivered. Within a national network different SSNs, according to Recommendation Q.713, may be allocated for the different network-specific applications, e.g. a SSN may be allocated for a SCF functionality.

Method b) above may be used when a message is to be delivered to a SCCP-user which can be identified by the combination of the elements within the GT. An example of the use of this method is when messages have to be delivered between different networks. This method may be used since the originating network is unaware of the point code and SSN's allocations within the destination network. The network that determines the end-node to which the message is to be delivered has to perform a Global Title Translation to derive the destination Point Code and the SSN. If optionally the original address contained the SSN, then this may be used as the destination SSN, or the translation may, if required, provide an appropriate new SSN. Where the destination node is in another network (and is not the gateway node) then the application populates the SSN field with either the SSN in use at the destination or zero.

When GT is used for addressing, the IN application expects that the SCCP supports the following elements as defined in Recommendation Q.713:

### **Global Title Indicator**

This indicator specifies the method employed for the formatting of the address information. The format with the indicator value 4 is always used for internetwork connections.

### **Translation Type**

The Translation Types are defined within Recommendation Q.713.

### **Numbering Plan**

- 1) For Addresses used on internetwork interfaces the Numbering Plan must be either Generic Numbering, or E.164 as defined in Annex B/Q.713.
- 2) For Addresses used on all other interfaces, any of the Numbering plans defined in Recommendation Q.713 may be used if deemed suitable.

### **Global Title Address Information**

This is the actual INAP address information supplied by the application and is encoded as indicated by the encoding scheme.

### **Encoding scheme**

The application should set the value of the encoding scheme according to the format of the GTAI. The allowed values are defined in Recommendation Q.713.

The network provider must ensure that any change of GT value during translation preserves any INAP specific information contained in the initial GT value. The GT translation data in the network must not delete the GT information, if present, from the Address.

This requirement applies to all interfaces, not just those used for internetworking.

If *route on SSN* is to be supported from the originating node, then a non-zero internationally standardised SSN is required for international internetworking.

In the absence of a standardized non-zero SSN for INAP services, the use of *route on GT* is mandatory from the origin node to the network containing the destination node.

The version of SCCP used to support INAP operations must be at least *White Book*, 1993.

#### **18.2.2.1.2 Sequence control**

The application will specify whether SCCP protocol class 0 or 1 is required.

Class 0 is used when the in-sequence delivery of messages to a specific called address is not required.

Class 1 is used when the in-sequence delivery of messages to a specific called address is required.

#### **18.2.2.1.3 Return on error**

The use of Return on Error mechanism may be required by the IN applications so that the application is aware of messages that have not been delivered to the destination by the SCCP. The return option allows the return of the message that was not delivered due to routing or segmentation/reassembly failure back to the issuing user.

If the return option is invoked by the application and the message is not delivered, then the SCCP specifies the "return reason" as specified in Recommendation Q.711. The N-NOTICE primitive is used to return the undelivered message to the originating user.

#### **18.2.2.1.4 Segmentation/reassembly**

The application expects that since the SCCP can send up to 260 octets of user data (including the address information and TC-message) in a UDT message (248 octets in a XUDT message performing segmentation and congestion control), segmentation is available for long user data.

Also the SCCP is expected to perform the reassembly function on received segmented messages and deliver the reassembled user data to the user.

However, it should be noted that even though the theoretical maximum size of SCCP-user data and addresses that can be segmented by the SCCP is 3968 octets, the SCCP-user would limit the length to about 2560 octets to allow for the largest known addresses. Note that the application must also allow for the octets used for the TC-message in the 2560 octets.

The IN application does not expect the SCCP to segment the user data into more than 16 segments.

#### **18.2.2.1.5 Congestion control**

To help control of possible congestion that might occur in the lower layers, the application may assign a value to indicate the importance of the message. The use of this parameter requires the use of SCCP [Q.711-Q.714 (07/96) and Q.715 (SCCP User Guide)] Recommendations.

Also there exist other congestion control mechanisms as indicated below in SCCP Management.

### **18.2.2.2 SCCP Connection-oriented Services**

The use by IN applications for the Connection-oriented services is outside the scope of IN CS-2.

#### **18.2.2.3 SCCP management**

The subsystems used within the IN scenario expect the SCCP to provide management procedures to maintain network performance by re-routing in the event of failure of a subsystem, and in case of network congestion by use of the congestion handling procedure. These procedures have appropriate interactions with the SCCP user as described in Recommendation Q.711.

To achieve the above, the SCCP is expected to perform the following procedures:

- Signalling point status management (N-PCSTATE, which includes the signalling point prohibited, signalling point allowed, signalling point congested, and local MTP availability subprocedures).

- Subsystem status management (N-STATE, which includes the subsystem prohibited, subsystem allowed, and subsystem status test subprocedures).
- Coordinated state change (N-COORD, which is a procedure which allows a duplicated subsystem to be withdrawn from service without affecting the performance of the network).

## **19 IN generic interface security**

Any interface within the IN functional architecture may have the need to apply security functions to the information flows passing across the interface. This clause defines a generic set of security mechanisms and procedures based on the X.509 Authentication Framework and the Simple Public Key GSS API Mechanism (SPKM) to enable any interfaces to provide suitable secured communications. This clause defines a sub-profile of X.509 and SPKM for use of IN CS-2.

For IN CS-2, the provision for security functions is required for the SCF-SDF, SDF-SDF, and SCF-SCF interfaces.

### **19.1 Interface security requirements**

The security requirements on an IN system could be divided in the two following main families:

- 1) Requirements to offer security features (Network access security requirements): This covers the various aspects relating to the protection of user access to services and of terminals to networks against attacks at the access interface (for instance user impersonation), by means of security features such as: user/terminal authentication (i.e. the result of a process by which a service user proves his or her identity to an IN system), user profile verification (i.e. the verification that a user is authorised to use a functionality), etc.
- 2) Requirements on the internetworking interfaces (Internetworking security requirements): This covers the protection of interactions between the various entities and agents involved in the provision of a telecommunication service against attacks at the internal interfaces of the system or even against data stored inside equipment, by means of security features such as: peer entity authentication (i.e. a process which allows a communicating entity to prove its identity to another entity in the network), signalling data or TMN data integrity, non-repudiation, confidentiality, entity profile verification (i.e. the verification that an entity is authorised to use a functionality), etc.

For each of the above problems, the interfaces have to provide processes to enable data confidentiality, data origin authentication and data integrity. These capabilities are generally described in Recommendation X.500.

#### **19.1.1 Data confidentiality**

Data Confidentiality ensures that only the parties involved can examine the contents of the message. Encryption, which applies an algorithm to transform the message into a protected form, is sufficient to provide data confidentiality. The encryption methods for all communications with the directory shall be conformant to Directory Security Mechanisms or to directory entry methods.

#### **19.1.2 Data integrity and data origin authentication**

Data Integrity ensures that the contents of the message have not been altered during transmission. Data Origin Authentication ensures that the message came from the expected origin. Digital Signatures, which applies an encryption algorithm to the result of applying a hash function to the contents of message, are sufficient for providing data integrity and data origin authentication. Data integrity methods for all communications with the directory shall be conformant to Directory Security Mechanisms or to directory entry methods.

### 19.1.3 Key management

Key Management refers to the generation, distribution and maintenance of private keys used in the cryptographic system. Key Management methods conformant with X.509 Recommendation shall be used.

## 19.2 Procedures and algorithms

Several assumptions are provided to the IN CS-2 interface security procedures to focus the work, but which also provides acceptable levels of security (see Table 19-1):

- Generic Security Interface shall be based on SPKM as much as possible.
- The number of parameters in messages and procedures should be minimized, because each of them adds to the signalling traffic load and to some processing time.
- Only one type of security shall be applied to the interface. The order of selection will be none, User, Peer:

**Table 19-1/Q.1228 – Categories of security levels**

	Signature	Encryption
None	None	None
User	None	User
	User	None
	User	User
Peer	None	Peer
	Peer	None
	Peer	Peer

- If both user and peer signature/encryption are requested, then user signature/encryption should be used.
- Only the mechanisms Encryption and Digital Signatures [including Message Authentication Code (MAC) techniques] shall be used.

### 19.2.1 Authentication procedures

The Authentication Procedure defines the process of establishing authentication and shall perform the following functions:

- Authenticate the Entity (User/FE) as being allowed to access an FE.
- Establish the level of security for all messages between the Entity and the FE.
- If required, establish a session key for encryption of messages between the Entity and the FE.

The current IN CS-2 authentication uses possible procedures defined in Recommendation X.509, together with a new procedure, SPKM. The primary purpose of the X.509 Recommendation is to provide authentication for a directory service, which includes mechanisms on Simple Authentication, Strong Authentication and SPKM. This subclause indicates which aspects of the Directory Authentication should be considered and supported by implementors within the scope of IN CS-2.

The current CS-2 authentication procedures are:

- 1) Simple Authentication (see 6.2/X.509), using Bind operations with passwords for IN CS-1 (1995) or IN CS-2.
- 2) Strong Authentication, (see clause 10/X.509), using Bind operations with strong credentials for IN CS-2 and 1993 Directory; or using SPKM for IN CS-2.
- 3) External Procedures, using Bind operations for IN CS-1 (1995) or IN CS-2.
- 4) SPKM (Simple Public Key GSS-API Mechanism):
  - The SPKM allows both unilateral and mutual authentication to be accomplished without the use of secure timestamps. This enables environments which do not have access to secure time to nevertheless have access to secure authentication.
  - The SPKM uses Algorithm Identifiers to specify various algorithms (for data confidentiality, data integrity, etc.) to be used by the communicating peers. This allows maximum flexibility for a variety of environments, for future enhancements, and for alternative algorithms.

### **19.2.2 SPKM algorithms and negotiation**

Additional information on SPKM is provided in this subclause. For additional information on other authentication mechanisms, including Directory entry methods, please refer to the user guide and relevant Directory specifications.

A number of algorithm types are employed in SPKM. Examples of algorithms are presented in Appendix III. It should be noted that in the SPKM algorithms, the terms "initiator" and "target" are used. For IN CS-2, the term "target" may be referred to as "consumer".

#### **19.2.2.1 Data Confidentiality Algorithm (C-ALG)**

The symmetric algorithm is used to generate the encrypted data. The DES-CBC algorithm is an example for data confidentiality.

#### **19.2.2.2 Data Integrity Algorithm (I-ALG)**

The purpose of this algorithm is to ensure that a message has not been altered in any way after being constructed by the legitimate sender. Depending on the algorithm used, the application of this algorithm may also provide authenticity and support non-repudiation (including data origin authentication) for the message. Algorithms such as md5WithRSA, DES-MAC, md5-DES-CBC, and sum64-DES-CBC are examples for data integrity.

#### **19.2.2.3 Key Management Algorithm (K-ALG)**

The purpose of this algorithm is to establish a symmetric key (called context key) for use by both the initiator and the responder (called target) over the established context. The keys used for C-ALG and any keyed I-ALGs (for example, DES-MAC) are derived from this context key. Key establishment is done within the X.509 authentication exchange and so the resulting shared symmetric key is authenticated. Algorithms such as RSAEncryption and dhKeyAgreement are examples for Key Management.



#### **19.2.2.4 One-Way Function (O-ALG) for Subkey Derivation Algorithm**

Having established a context key using the negotiated K-ALG, both initiator and target must be able to derive a set of subkeys for the various C-ALGs and keyed I-ALGs.

#### **19.2.2.5 SPKM negotiation**

During context establishment in SPKM, the initiator offers a set of possible confidentiality algorithms and a set of possible integrity algorithms to the target (note that the term "integrity algorithms" includes digital signature algorithms). The confidentiality algorithms selected by the target become those that may be used for C-ALG over the established context, and the integrity algorithms selected by the target become those that may be used for I-ALG over the established context (the target "selects" algorithms by returning, in the same relative order, the subset of each offered list that it supports). Note that any C-ALG and I-ALG may be used for any message over the context and that the first confidentiality algorithm and the first integrity algorithm in the agreed sets become the default algorithms for that context.

The agreed confidentiality and integrity algorithms for a specific context define the valid values of the Quality of Protection (QOP) parameter. If no response is expected from the target in unilateral authentication, then the algorithms offered by the initiator are the ones that may be used over the context (if this is unacceptable to the target, then the association with this context will not be successfully established).

Furthermore, in the first context establishment token the initiator offers a set of possible K-ALGs, along with the key (or key half) corresponding to the first algorithm in the set (its preferred algorithm). If this K-ALG is unacceptable to the target, then the target must choose one of the other K-ALGs in the set and send, this choice along with the key (or key half) corresponding to this choice in its response (otherwise the association with this context will not be established). If necessary (that is, if the target chooses a 2-pass K-ALG such as dhKeyAgreement), the initiator will send its key half in a response to the target.

Finally, in the first context establishment token the initiator offers a set of possible O-ALGs (subkey derivation algorithms) (only a single O-ALG if no response is expected). The (single) O-ALG chosen by the target becomes the subkey derivation algorithm to be used over the context.

### **19.2.3 Three-way mutual authentication**

The use of three-way mutual authentication procedures (including SPKM) in the establishment of a security association between two Functional Entities (FEs) implies the use of the Generic Upper Layers Security (GULS) Security Exchange Service Element (SESE) protocol (defined in Recommendations X.831 and X.832) as described in Recommendation X.519.

#### **19.2.4 Assignment of credentials**

Procedures for assigning credentials to users or FEs are generally defined in Recommendation X.500.

### **19.3 Mapping of security information flow definitions to tokens**

In the case of the SCF-SDF interface, the Context Establishment Tokens will be conveyed in Bind procedure and the Per-Message Tokens will be used in DAP operations.

In the case of the SDF-SDF interface, the Context Establishment Tokens will be conveyed in Bind procedure and the Per-Message Tokens will be used in DSP and DISP operations.

In the case of the SCF-SCF interface, the Context Establishment Tokens will be conveyed in the SCF Bind procedure and the Per-Message Tokens will be used in SCF-SCF operations and Distributed SCF System operations.

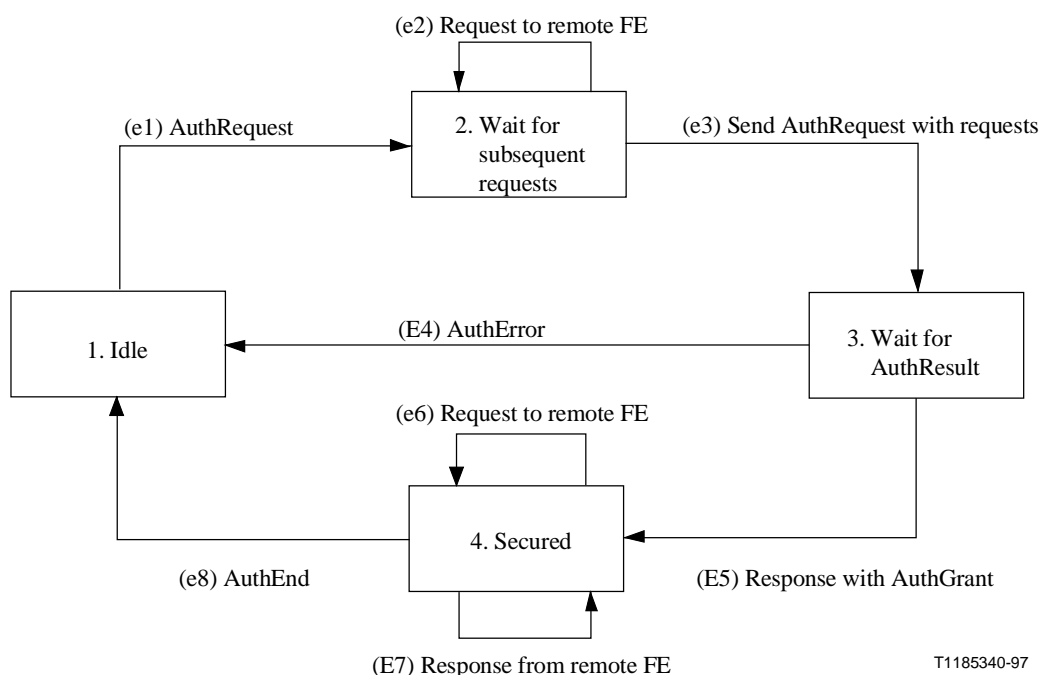
## 19.4 Security FSM definitions

The Security FSMs described here are generic machines. They illustrate how the security procedures are handled across the interface between two FEs.

### 19.4.1 Two-way mutual authentication FSMs

#### 19.4.1.1 Outgoing FSM

The outgoing FSM controls the originating side of the interface. It is shown in Figure 19-1.



**Figure 19-1/Q.1228 – The originating FSM states**

##### 19.4.1.1.1 State 1: Idle

The following event is considered in this state:

- (e1) **AuthRequest**: This is an internal event, caused by the need of the service logic to create an association with a remote FE in order to begin accessing a service function. This event causes a transition to the state 2, **Wait for subsequent requests**.

##### 19.4.1.1.2 State 2: Wait for subsequent requests

In this state, subsequent operations to be sent with the **AuthRequest** operation (in the same message) to the remote FE are expected. The following two events are considered in this state:

- (e2) **Request\_to\_remote\_FE**: This is an internal event caused by the reception of an operation. The operation is buffered until the reception of a delimiter (or a timer expiration). The Originating FSM remains in the same state; and

- (e3) **Send\_AuthRequest\_with\_requests**: This is an internal event caused by the reception of a delimiter, that indicates the reception of the last operation to be sent. Once the delimiter is received, a message containing the argument of the **AuthRequest** operation and other operations' arguments, if any, is sent to the remote FE. This event causes a transition out of this state to the state 3, **Wait for AuthResult**.

#### 19.4.1.1.3 State 3: Wait for AuthResult

In this state, the local FE is waiting for the response from the remote FE. Two events are considered in this state:

- (E4) **Auth\_Error**: This is an external event, caused by the reception of an error to the **AuthRequest** operation previously issued to the remote FE. This event causes a transition out of this state to state 1, **Idle**; and
- (E5) **Response\_with\_AuthGrant**: This is an external event, caused by the reception of a **AuthRequest** result combined with the responses to other operations previously issued to the remote FE (if any). This event causes a transition to state 4, **Secured**.

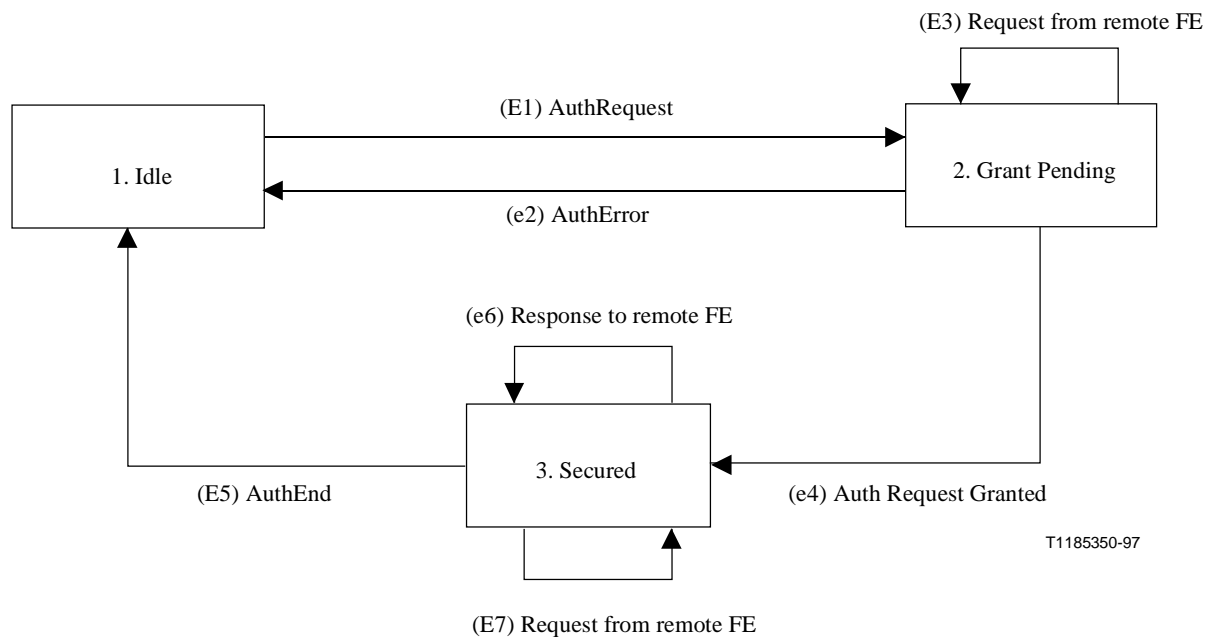
#### 19.4.1.1.4 State 4: Secured

In this state, the local FE has established an authenticated access to the remote FE, and is waiting for requests to the remote FE from the service logic or is waiting for responses to the operations previously issued to the remote FE. Three events are considered in this state:

- (e6) **Request\_to\_remote\_FE**: This is an internal event, caused by the need of the service logic to access the remote FE. The Originating FSM remains in the same state;
- (E7) **Response\_from\_remote\_FE**: This is an external event, caused by the reception of responses to the operations previously issued to the remote FE. The Originating FSM remains in the same state; and
- (e8) **AuthEnd**: This is an internal event, caused by the need of the service logic to terminate the authenticated access to the remote FE. This event causes a transition state to state 1, **Idle**.

#### 19.4.1.2 Incoming FSM

The incoming FSM controls the terminating side of the interface. It is shown in Figure 19-2.



**Figure 19-2/Q.1228 – The terminating FSM states**

Each state is discussed in one of the following subclauses.

General rules applicable to more than one state are as follows:

In any state, if the dialogue with the Remote FE is terminated, then the Terminating FSM returns to **Idle** state after ensuring that any resources allocated to the call have been de-allocated.

#### 19.4.1.2.1 State 1: Idle

The only event accepted in this state is:

- (E1) AuthRequest: This is an external event caused by the reception of AuthRequest operation from the remote FE. This event causes a transition out of this state to state 2, **Grant Pending**.

#### 19.4.1.2.2 State 2: Grant Pending

In this state, an Auth Request has been received from the remote FE. The local FE is performing the authentication procedures. Three events are considered in this state:

- (e2) AuthError: This is an internal event, caused by the failure of the AuthRequest operation previously issued to the local FE. This event causes a transition out of this state to state 1, **Idle** and a AuthError is returned to the invoking SCF;
- (E3) Request\_from\_remote\_FE: This is an external event, caused by the reception of operations before the result from the AuthRequest operation is determined. The SDSM remains in the same state; and
- (e4) AuthRequest\_Granted: This is an internal event, caused by the successful completion of the AuthRequest operation previously issued to the local FE. This event causes a transition out of this state to state 3, **Secured**.

### 19.4.1.2.3 State 3: Secured

In this state, the access of the remote FE to the local FE was authorized and operations coming from the remote FE are accepted. Besides waiting for requests from the remote FE, the local FE can send responses to previously issued operations. Three events are considered in this state:

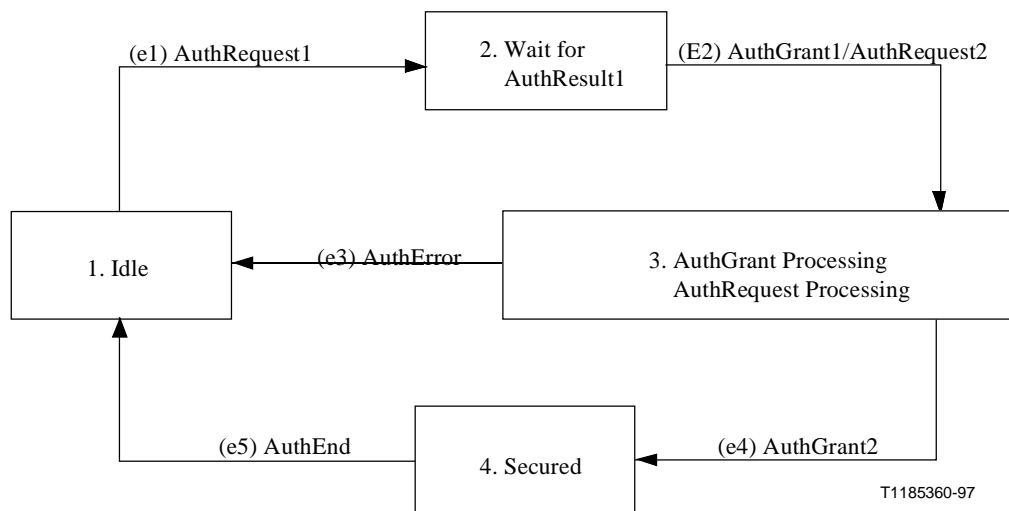
- (E5) AuthEnd: This is an external event, caused by the reception of the AuthEnd operation from the remote FE. The authorised association is ended and all associated resources are released. This event causes a transition out of this state to state 1, **Idle**;
- (e6) Response\_to\_remote\_FE: This is an internal event, caused by the completion of the operations previously issued by the remote FE. The results or errors of these operations are returned to the remote FE. The Terminating FSM remains in the same state; and
- (E7) Request\_from\_remote\_FE: This is an external event, caused by the reception of a request from the remote FE. The Terminating FSM remains in the same state.

### 19.4.2 Three-way mutual authentication FSMs

An outgoing and incoming generic security FSMs for three-way mutual authentication are described in this subclause. The Security FSMs described here are generic machines. They illustrate how the security procedures are handled across the interface between two FEs.

#### 19.4.2.1 Outgoing FSM for three-way mutual authentication

The outgoing FSM controls the originating side of the interface. It is shown in Figure 19-3.



**Figure 19-3/Q.1228 – The originating FSM states for three-way mutual authentication**

#### 19.4.2.1.1 State 1: Idle

The following event is considered in this state:

- (e1) AuthRequest1: This is an internal event, caused by the need of the service logic to create an association with a Remote FE in order to begin accessing a service function. This event causes a transition to the state 2, **Wait for AuthResult1**.

#### 19.4.2.1.2 State 2: Wait for AuthResult1

In this state, a response to the initial challenge (sent in the AuthRequest1) from the remote FE is expected. The following event is considered in this state:

- (E2) AuthGrant1/AuthRequest2: This is an external event caused by the reception of a response to the initial challenge and a subsequent challenge from the remote FE. This event causes a transition out of this state to the state 3, **AuthGrant Processing/AuthRequest Processing**.

#### 19.4.2.1.3 State 3: AuthGrant Processing/AuthRequest Processing

In this state, the local FE confirms the correct response to the AuthRequest1. In addition, an appropriate response to AuthRequest2 is created. Two events are considered in this state:

- (e3) AuthError: This is an internal event, caused by the an error in processing AuthGrant1 received from the remote FE in response to **AuthRequest1**. This event causes a transition out of this state to state 1, **Idle**; and
- (e4) AuthGrant2: This is an internal event, caused by the processing of an appropriate response to AuthRequest2 which is issued to the remote FE. This event causes a transition to state 4, **Secured**.

#### 19.4.2.1.4 State 4: Secured

In this state, the local FE has established a unilateral authentication with the remote FE. Once the remote FE also progresses to a Secured state, requests to and from the remote FE may be processed (these events are not shown for clarification). One event is considered in this state:

- (e5) AuthEnd: This is an internal event, caused by the need of the service logic to terminate the authenticated access to the remote FE. This event causes a transition state to state 1, **Idle**.

#### 19.4.2.2 Incoming FSM for three-way mutual authentication

The incoming FSM controls the terminating side of the interface. It is shown in Figure 19-4.

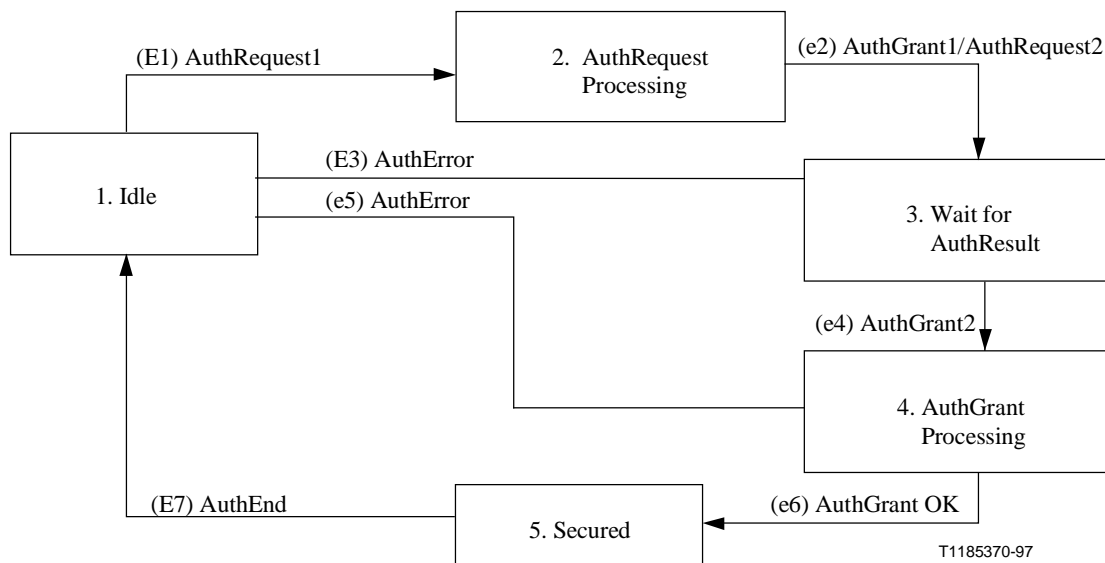


Figure 19-4/Q.1228 – The terminating FSM states for three-way mutual authentication

#### 19.4.2.2.1 State 1: Idle

The only event accepted in this state is:

- (E1) AuthRequest1: This is an external event caused by the reception of AuthRequest1 information from the remote FE. This event causes a transition out of this state to state 2, **AuthRequest Processing**.

#### 19.4.2.2.2 State 2: AuthRequest Processing

In this state, an AuthRequest has been received from the remote FE. The local FE is performing the authentication procedures. One event is considered in this state:

- (e2) AuthGrant1/AuthRequest2: This is an internal event, caused by the successful completion of the AuthRequest1 operation previously issued to the local FE. In addition, the local FE authentication challenge (AuthRequest2) is generated. This event causes a transition out of this state to state 3, **Wait for AuthResult**.

#### 19.4.2.2.3 State 3: Wait for AuthResult

In this state, the local FE awaits confirmation from the remote FE that either the AuthGrant1 response to AuthRequest1 challenge was accepted, or a response (AuthGrant2) to the AuthRequest2 challenge is received. Two external events are considered in this state:

- (E3) AuthError: This is an external error event, generated by the remote FE, on receipt of an invalid response to its challenge. This event causes a transition out of this state to state 1, **Idle**; and
- (E4) AuthGrant2: This is an external event, generated by the remote FE, containing the response (AuthGrant2) to the local FE challenge (AuthRequest2). This event causes a transition out of this state to state 4, **AuthGrant Processing**.

#### 19.4.2.2.4 State 4: AuthGrant Processing

In this state, the local FE confirms the received response (AuthGrant2) to the challenge (AuthRequest2). Two internal events are considered in this state:

- (e5) AuthError: This is an internal error event, generated by the local FE, on receipt of an invalid response to its challenge. This event causes a transition out of this state to state 1, **Idle**; and
- (e6) AuthGrant OK: This is an internal event, generated by the local FE, on receipt of a valid response to its challenge. This event causes a transition to state 5, **Secured**.

#### 19.4.2.2.5 State 5: Secured

In this state, the local FE has established a unilateral authentication with the remote FE. Since the remote FE is also in a Secured state, requests to and from the remote FE may be processed (these events are not shown for clarification). One event is considered in this state:

- (E7) AuthEnd: This is an external event, caused by the need of the service logic to terminate the authenticated access to the remote FE. This event causes a transition state to state 1, **Idle**.





## ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
<b>Series Q</b>	<b>Switching and signalling</b>
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure
Series Z	Programming languages