

**RAPPORT
TECHNIQUE
TECHNICAL
REPORT**

**CEI
IEC**

60870-6-504

Première édition
First edition
1998-12

Matériels et systèmes de téléconduite –

Partie 6-504:

**Protocoles de téléconduite compatibles avec les
normes ISO et les recommandations de l'UIT-T –
Conventions d'utilisation de TASE.1**

Telecontrol equipment and systems –

Part 6-504:

**Telecontrol protocols compatible with ISO
standards and ITU-T recommendations –
TASE.1 user conventions**



Numéro de référence
Reference number
CEI/IEC 60870-6-504:1998

Numéros des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60000.

Publications consolidées

Les versions consolidées de certaines publications de la CEI incorporant les amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles dans le Catalogue de la CEI.

Les renseignements relatifs à des questions à l'étude et des travaux en cours entrepris par le comité technique qui a établi cette publication, ainsi que la liste des publications établies, se trouvent dans les documents ci-dessous:

- «Site web» de la CEI*
- **Catalogue des publications de la CEI**
Publié annuellement et mis à jour régulièrement (Catalogue en ligne)*
- **Bulletin de la CEI**
Disponible à la fois au «site web» de la CEI* et comme périodique imprimé

Terminologie, symboles graphiques et littéraux

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 60050: *Vocabulaire Electrotechnique International* (VEI).

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera la CEI 60027: *Symboles littéraux à utiliser en électrotechnique*, la CEI 60417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles*, et la CEI 60617: *Symboles graphiques pour schémas*.

* Voir adresse «site web» sur la page de titre.

Numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series.

Consolidated publications

Consolidated versions of some IEC publications including amendments are available. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available in the IEC catalogue.

Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is to be found at the following IEC sources:

- **IEC web site***
- **Catalogue of IEC publications**
Published yearly with regular updates (On-line catalogue)*
- **IEC Bulletin**
Available both at the IEC web site* and as a printed periodical

Terminology, graphical and letter symbols

For general terminology, readers are referred to IEC 60050: *International Electrotechnical Vocabulary* (IEV).

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications IEC 60027: *Letter symbols to be used in electrical technology*, IEC 60417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets* and IEC 60617: *Graphical symbols for diagrams*.

* See web site address on title page.

RAPPORT
TECHNIQUE – TYPE 2

CEI
IEC

TECHNICAL
REPORT – TYPE 2

60870-6-504

Première édition
First edition
1998-12

Matériels et systèmes de téléconduite –

Partie 6-504:

**Protocoles de téléconduite compatibles avec les
normes ISO et les recommandations de l'UIT-T –
Conventions d'utilisation de TASE.1**

Telecontrol equipment and systems –

Part 6-504:

**Telecontrol protocols compatible with ISO
standards and ITU-T recommendations –
TASE.1 user conventions**

© IEC 1998 Droits de reproduction réservés — Copyright - all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

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 publisher.

International Electrotechnical Commission
Telefax: +41 22 919 0300

3, rue de Varembé Geneva, Switzerland
e-mail: inmail@iec.ch IEC web site <http://www.iec.ch>



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE

XH

Pour prix, voir catalogue en vigueur
For price, see current catalogue

SOMMAIRE

	Pages
AVANT-PROPOS	6
INTRODUCTION	10
Articles	
1 Domaine d'application et objet	12
2 Références normatives	12
3 Définitions et abréviations	14
3.1 Définitions	14
3.2 Abréviations	20
4 Élément utilisateur de TASE.1	22
4.1 Modèle conceptuel	22
4.2 Unités fonctionnelles (FU)	24
4.3 Structure commune des descriptions de FU	30
5 Function group gestion d'association	32
5.1 Adressage	32
5.2 Partie adresse	34
5.3 Marquage de redémarrage	38
5.4 Contrôle d'intégrité de la configuration de groupe	38
5.5 Permanent Association FU	42
5.6 Dynamic Association FU	54
5.7 Test Association FU	70
6 Function group identification de données	78
6.1 Attributs de groupe	78
6.2 Group Management FU	88
6.3 Group Definition FU	106
6.4 Group Readout FU	126
6.5 Group Configuration FU	146
7 Function group transfert de données	152
7.1 Requested Data Transfer FU	152
7.2 Periodically Requested Data Transfer FU	166
7.3 Unsolicited Data Transfer FU	172
7.4 Unsolicited Periodic Data Transfer FU	186
7.5 Unsolicited Spontaneous Data Transfer FU	204
7.6 Supervisory Control Data Transfer FU	222
8 Function group redémarrage	242
8.1 Restart Reconfigure FU	242
8.2 Restart Reactivate FU	252

CONTENTS

	Page
FOREWORD	7
INTRODUCTION	11
Clause	
1 Scope and object	13
2 Normative references	13
3 Definitions and abbreviations.....	15
3.1 Definitions.....	15
3.2 Abbreviations	21
4 The TASE.1 User Element.....	23
4.1 Conceptual model	23
4.2 Functional units (FUs)	25
4.3 Common template for the descriptions of FUs	31
5 The Association Management Function Group	33
5.1 Addressing.....	33
5.2 Address part	35
5.3 Restart Marking	39
5.4 Group Configuration integrity control.....	39
5.5 Permanent Association FU	43
5.6 Dynamic Association FU.....	55
5.7 Test Association FU	71
6 The Data Identification Function Group.....	79
6.1 Group attributes	79
6.2 Group Management FU	89
6.3 Group Definition FU	107
6.4 Group Readout FU	127
6.5 Group Configuration FU	147
7 The Data Transfer Function Group	153
7.1 Requested Data Transfer FU.....	153
7.2 Periodically Requested Data Transfer FU	167
7.3 Unsolicited Data Transfer FU	173
7.4 Unsolicited Periodic Data Transfer FU.....	187
7.5 Unsolicited Spontaneous Data Transfer FU	205
7.6 Supervisory Control Data Transfer FU	223
8 The Restart Function Group	243
8.1 Restart Reconfigure FU.....	243
8.2 Restart Reactivate FU	253

9	Sécurité	260
9.1	Introduction.....	260
9.2	Services de sécurité.....	260
9.3	Mécanismes de sécurité.....	262
9.4	Classes de sécurité et options.....	264
9.5	Conseils d'utilisation des mécanismes de sécurité.....	264
9.6	Définition de la zone d'information de sécurité.....	266
Annexe A (normative)	Hiérarchie d'invocation des FU	270
Annexe B (normative)	Hiérarchie d'interruption des FU	274
Annexe C (normative)	Utilisation des valeurs de code de résultat	278
Annexe D (informative)	Bibliographie	282
Annexe E (normative)	Proforma UECS pour les éléments utilisateurs TASE.1	284
Annexe F (normative)	Proforma UEXIT pour les éléments utilisateurs TASE.1	332

IECNORM.COM : Click to view the full PDF of IEC TS 60870-6-504:1998

9	Security.....	261
9.1	Introduction.....	261
9.2	Security services.....	261
9.3	Security mechanisms	263
9.4	Security classes and options	265
9.5	Suggested use of the security mechanisms	265
9.6	Definition of the security information field	267
Annex A (normative)	FU invocation hierarchy.....	271
Annex B (normative)	FU disruption hierarchy	275
Annex C (normative)	Use of result code values	279
Annex D (informative)	Bibliography	283
Annex E (normative)	UECS proforma for TASE.1 User Elements	285
Annex F (normative)	UEXIT proforma for the TASE.1 User Elements.....	333

IECNORM.COM : Click to view the full PDF of IEC TS 60870-6-504:1998

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

MATÉRIELS ET SYSTÈMES DE TÉLÉCONDUITE –

Partie 6-504: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Conventions d'utilisation de TASE.1

AVANT-PROPOS

- 1) La CEI (Commission Electrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI, entre autres activités, publie des Normes internationales. Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible un accord international sur les sujets étudiés, étant donné que les Comités nationaux intéressés sont représentés dans chaque comité d'études.
- 3) Les documents produits se présentent sous la forme de recommandations internationales. Ils sont publiés comme normes, rapports techniques ou guides et agréés comme tels par les Comités nationaux.
- 4) Dans le but d'encourager l'unification internationale, les Comités nationaux de la CEI s'engagent à appliquer de façon transparente, dans toute la mesure possible, les Normes internationales de la CEI dans leurs normes nationales et régionales. Toute divergence entre la norme de la CEI et la norme nationale ou régionale correspondante doit être indiquée en termes clairs dans cette dernière.
- 5) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand un matériel est déclaré conforme à l'une de ses normes.
- 6) L'attention est attirée sur le fait que certains des éléments de la présente Norme internationale peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

La tâche principale des comités d'études de la CEI est d'élaborer des Normes internationales. Exceptionnellement, un comité d'études peut proposer la publication d'un rapport technique de l'un des types suivants:

- type 1, lorsque, en dépit de maints efforts, l'accord requis ne peut être réalisé en faveur de la publication d'une Norme internationale;
- type 2, lorsque le sujet en question est encore en cours de développement technique ou lorsque, pour une raison quelconque, la possibilité d'un accord pour la publication d'une Norme internationale peut être envisagée pour l'avenir mais pas dans l'immédiat;
- type 3, lorsqu'un comité d'étude a réuni des données de nature différente de celles qui sont normalement publiées comme Normes internationales, cela pouvant comprendre, par exemple des informations sur l'état de la technique.

Les rapports techniques des types 1 et 2 font l'objet d'un nouvel examen trois ans au plus tard après leur publication afin de décider éventuellement de leur transformation en Normes internationales. Les rapports techniques du type 3 ne doivent pas nécessairement être révisés avant que les données qu'ils contiennent ne soient plus jugées valables ou utiles.

La CEI 60870-6-504, rapport technique de type 2, a été établie par le comité d'études 57 de la CEI: Conduite des systèmes de puissance et communications associées.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

TELECONTROL EQUIPMENT AND SYSTEMS –

**Part 6-504: Telecontrol protocols compatible with ISO standards
and ITU-T recommendations – TASE.1 user conventions**

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but no immediate possibility of an agreement on an International Standard;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

Technical reports of types 1 and 2 are subject to review within three years of publication to decide whether they can be transformed into International Standards. Technical reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

IEC 60870-6-504, which is a technical report of type 2, has been prepared by IEC technical committee 57: Power system control and associated communications.

Le texte de ce rapport technique est issu des documents suivants:

Projet de comité	Rapport de vote
57/282/CDV	57/331/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de ce rapport technique.

Le présent document est publié dans la série des rapports techniques de type 2 (conformément au G.3.2.2 de la partie 1 des Directives CEI/ISO) comme «norme prospective d'application provisoire» dans le domaine des matériels et systèmes de téléconduite car il est urgent d'avoir des indications sur la meilleure façon d'utiliser les normes dans ce domaine afin de répondre à un besoin déterminé.

Ce rapport ne doit pas être considéré comme une «Norme internationale». Il est proposé pour une mise en oeuvre provisoire, dans le but de recueillir des informations et d'acquérir de l'expérience quant à son application dans la pratique. Il est de règle d'envoyer les observations éventuelles relatives au contenu de ce rapport au Bureau Central de la CEI.

Il sera procédé à un nouvel examen de ce rapport technique de type 2 trois ans au plus tard après sa publication, avec la faculté d'en prolonger la validité pendant trois autres années, de le transformer en Norme internationale ou de l'annuler.

Les annexes A, B, C, E et F font partie intégrante de ce rapport technique.

L'annexe D est donnée uniquement à titre d'information.

IECNORM.COM : Click to view the full PDF of IEC 60870-6-504:1998

The text of this technical report is based on the following documents :

Committee draft	Report on voting
57/282/CDV	57/331/RVC

Full information on the voting for the approval of this technical report can be found in the report on voting indicated in the above table.

This document is issued in the type 2 technical report series of publications (according to G.3.2.2 of part 1 of the IEC/ISO Directives) as a “prospective standard for provisional application” in the field of telecontrol equipment and systems because there is an urgent requirement for guidance on how standards in this field should be used to meet an identified need.

This document is not to be regarded as an “International Standard”. It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the IEC Central Office.

A review of this type 2 of technical report will be carried out not later than three years after its publication, with the options of either extension for a further three years or conversion to an International Standard or withdrawal.

Annexes A, B, C, E and F form an integral part of this technical report.

Annex D is for information only.

IECNORM.COM : Click to view the full PDF of IEC 60870-6-504:1998

INTRODUCTION

TASE.1 Application Programming Interface (TAPI) (voir l'annexe A de la CEI 60870-6-501) fournit un ensemble de fonctions bien définies, qui permettent aux programmeurs de baser un processus d'application sur des protocoles de téléconduite pour acquérir des données en temps réel et contrôler des valeurs de consigne. La spécification TAPI n'assure cependant pas par elle-même une intégration fonctionnelle complète entre deux processus d'application de téléconduite.

Le présent rapport technique se fixe un double objectif:

- a) Fournir un jeu cohérent de règles applicables dans le cadre de l'utilisation de la TAPI, et, de manière plus précise, définir des règles pour:
 - utiliser des valeurs des paramètres TAPI;
 - mettre en séquence des primitives de service TASE (Telecontrol Application Service Element = Élément de Service d'Application pour la Téléconduite).
- b) Servir d'ouvrage de référence général pour définir les capacités des processus d'application de téléconduite. A cet effet, ce rapport présente divers paquets de fonctions «élémentaires», ou unités fonctionnelles (Functional Units: FU). Pour qu'un processus d'application de téléconduite soit déclaré conforme à ce rapport, il convient que ses caractéristiques interopératoires soient définies, en procédant à une énumération des unités fonctionnelles (FU) qu'il supporte.

La structure du paramètre «Data» de la primitive de transfert de données et du paramètre de données utilisateur de la primitive d'établissement de l'association sont décrites dans l'annexe A de la CEI 60870-6-501. Ces descriptions sont accompagnées d'informations liées à ces paramètres: types de données, codes de qualité, etc.

L'annexe E fournit un Proforma UECS pour les éléments utilisateurs TASE.1 qui constitue une «attestation de conformité d'un élément utilisateur» (UECS).

L'annexe F fournit un Proforma UEXIT qui donne des «informations supplémentaires sur l'élément utilisateur pour les tests» (UEXIT).

INTRODUCTION

The TASE.1 Application Programming Interface (TAPI) (see annex A of IEC 60870-6-501) provides a set of well-defined functions to an application programmer wishing to base an application process on the Telecontrol protocols for real-time data acquisition and setpoint control. However, the TAPI specification does not by itself ensure full functional integration between any two Telecontrol application processes.

The purpose of this technical report is twofold:

- a) To provide a consistent set of rules for the use of the TAPI, or more specifically, rules for:
 - usage of TAPI parameter values;
 - sequencing of Telecontrol Application Service Element service primitives.
- b) To be used as a general reference for specifying the capabilities of Telecontrol application processes. In order to achieve this, a number of "atomic" function packets, or Functional Units (FU), are defined: the interoperational characteristics of any Telecontrol application process claiming conformity with this document should be specified by quoting the Functional Units (FU) it supports.

The structure of the "Data" parameter of the data transfer primitives and the user data parameter in the association establishing primitive are described in annex A of IEC 60870-6-501, together with related information: data types, quality codes etc.

Annex E gives a UECS Proforma for the TASE.1 User Elements, which constitutes a User Element Conformance Statement (UECS).

Annex F gives a UEXIT Proforma, which provides a User Element Extra Information for Testing (UEXIT).

MATÉRIELS ET SYSTÈMES DE TÉLÉCONDUITE –

Partie 6-504: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Conventions d'utilisation de TASE.1

1 Domaine d'application et objet

Le présent rapport technique définit les règles applicables dans le cadre de l'utilisation de TASE.1 Application Programming Interface (TAPI), c'est-à-dire l'utilisation des paramètres et la mise en séquence des primitives de service.

Les règles contenues dans ce rapport technique sont à considérer comme une extension des règles définies par TASE.1. Elles limitent de façon encore plus précise les valeurs de paramètres, etc. Les limites et règles propres à TASE.1 sont également applicables, bien qu'il n'en soit pas explicitement fait état dans le présent rapport technique¹⁾. Lorsque le présent rapport technique autorise des invocations simultanées d'unités fonctionnelles (Functional Units: FU) sur une même association, le respect des règles de mise en séquence des primitives TASE.1²⁾ est implicite.

La structure des FU internes, de la fonction de coordination (voir article 4) et les relations d'ordre structurel entre celles-ci ne font pas partie du domaine d'application du présent rapport technique. Il s'agit en effet uniquement d'entités conceptuelles qui permettent de décrire plus facilement les séquences de primitives de service TASE.1.

Le domaine d'application du présent rapport technique n'inclut pas non plus les aspects non communicationnels des processus d'application, tels que la définition d'une base de données locale et l'accès à celle-ci, ou encore les éventuelles considérations de cohérence de données sur les différents modes d'accès aux données locales.

Le principal objet du présent rapport technique est la présentation du modèle conceptuel sous-jacent du Telecontrol User Element et l'identification des unités fonctionnelles (article 4). Les unités fonctionnelles (FU) sont décrites de façon très détaillée (articles 5 à 8) en présentant un groupe de fonctions par article.

2 Références normatives

Les documents normatifs suivants contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour le présent Rapport technique. Pour les références datées, les amendements ultérieurs ou les révisions de ces publications ne s'appliquent pas. Toutefois, les parties prenantes aux accords fondés sur le présent Rapport technique sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des documents normatifs indiqués ci-après. Pour les références non datées, la dernière édition du document normatif en référence s'applique. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur.

¹⁾ En conséquence, le traitement des erreurs de TAPI purement locales ne fait pas partie du domaine d'application de ce rapport technique. Les primitives de service TASE.1 de type *Req* et *Resp* sont toujours supposées être émises dans un contexte légal et avoir des valeurs de paramètres légales lorsqu'elles sont utilisées dans le cadre de TAPI. Pour obtenir des informations sur le cas plus spécial d'états d'appel local indiquant des problèmes de contrôle de flux (valeur d'état = -1), voir 4.2.1.

²⁾ Les règles de mise en séquence de primitives TASE.1 sont expliquées à l'article 8 de la CEI 60870-6-501. Elles sont exprimées sous forme d'un ensemble de transitions d'état légal à l'article 7 de la CEI 60870-6-502.

TELECONTROL EQUIPMENT AND SYSTEMS –

Part 6-504: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.1 user conventions

1 Scope and object

This technical report defines rules for usage of the TASE.1 Application Programming Interface (TAPI), i.e. parameter usage and service primitive sequencing.

The rules which are set forth by this technical report are to be considered as an extension to the rules enforced by the Telecontrol Application Service Element 1 (TASE.1) itself, and will typically constitute further restrictions on parameter values, etc. The TASE.1 own restrictions and rules apply throughout. They are not explicitly restated in this technical report¹⁾. Whenever the present report allows simultaneous Functional Unit invocations on one association, adherence to the TASE.1 primitive sequencing rules²⁾ are implicitly assumed.

Internal Functional Unit structures, Coordinating Function structure (see clause 4) and structural relationships between these are not within the scope of this technical report, as these are only conceptual entities, serving as a convenient means for describing TASE.1 service primitive sequences.

Also not within scope are all non-communicational aspects of application processes, such as local database design/access, and possible data consistency considerations across different local data access modes.

The main object of this technical report is to present the underlying conceptual model of the Telecontrol User Element and to identify the functional units (clause 4). The functional units are described in minute detail, grouped into one function group per clause (clauses 5 to 8).

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this technical report. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this technical report are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

¹⁾ Consequently, handling of purely local TAPI errors are outside the scope of this technical report. All specified TASE.1 service primitives of types *Req* and *Resp* are always assumed to be issued within legal context and have legal parameter values, as far as the TAPI is concerned. For the special case of local call status indicating flow control problems (status value = –1), see 4.2.1.

²⁾ The TASE.1 primitive sequencing rules are embodied in clause 8 of IEC 60870-6-501, and expressed as a set of legal state transitions in clause 7 of IEC 60870-6-502.

CEI 60870-6-1:1995, *Matériels et systèmes de téléconduite – Partie 6: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Section 1: Contexte applicatif et organisation des normes*

CEI 60870-6-501:1995, *Matériels et systèmes de téléconduite – Partie 6: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Section 501: Définitions des services TASE.1*

CEI 60870-6-502:1995, *Matériels et systèmes de téléconduite – Partie 6: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Section 502: Définitions du protocole TASE.1*

ISO 7498-2:1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 2: Architecture de sécurité*

UIT-T Recommandation X.509:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Annuaire: Cadre d'authentification*

3 Définitions et abréviations

3.1 Définitions

Pour les besoins du présent rapport technique, les définitions suivantes s'appliquent:

ACCEPTOR address

identification unique d'un User Element répondant à une tentative d'association. Voir 5.1 et 5.2.1

Changing a group

modification d'une ou plusieurs valeurs d'attribut du descripteur pour une identité de groupe existante

Composite FU

des composites FU agissent uniquement via l'invocation d'autres FU par un Initiator UE

Configuration Set

base de données de configuration de groupe partagée d'un commun accord par plusieurs systèmes INITIATOR/RESPONDER

Configuring a group

création et définition d'un groupe

Congestion Error

situation d'erreur dans laquelle TASE.1 n'est pas en mesure de recevoir de primitive de service de type *Req* ou *Resp* en raison d'un trafic important

NOTE – Le paragraphe 4.2.1 présente des règles générales de traitement des erreurs dues à un encombrement. Des règles spécifiques à chaque FU sont données dans chaque description de FU.

Coordinating Function

fonction User Element contrôlant les invocations de la Functional Unit locale

IEC 60870-6-1:1995, *Telecontrol equipment and systems – Part 6: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – Section 1: Application context and organization of standards*

IEC 60870-6-501:1995, *Telecontrol equipment and systems – Part 6: Telecontrol protocols compatible with ISO standards, and ITU-T recommendations – Section 501: TASE.1 Service definitions*

IEC 60870-6-502:1995, *Telecontrol equipment and systems – Part 6: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – Section 502: TASE.1 Protocol definitions*

ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection Basic Reference Model – Part 2: Security architecture*

ITU-T Recommendation X.509: 1993, *Information technology – Open systems interconnection – The directory: Authentication framework.*

3 Definitions and abbreviations

3.1 Definitions

For the purpose of this technical report, the following definitions apply:

ACCEPTOR address

the unique identification of a User Element responding to an Association attempt, see 5.1 and 5.2.1

Changing a group

modifying one or more of the descriptor attribute values for an existing group identity

Composite FU

composite FUs act via invocation of other FUs by an Initiator UE only. They have no associated specific TASE.1 Service primitive sequence. Neither do they have any specific RESPONDER part

Configuration Set

the currently agreed-upon group configuration database shared between a number of INITIATOR/RESPONDER systems

Configuring a group

creating and defining a group

Congestion error

error situation in which the TASE.1 is not able to receive *Req* or *Resp* type service primitives, because of heavy traffic

NOTE – General rules for handling congestion errors are given in 4.2.1. Special rules per FU are given in the individual FU descriptions.

Coordinating Function

User Element function that controls the local Functional Unit invocations

Creating a group

création d'une nouvelle identité de groupe³⁾ légale, en allouant un nouveau descripteur de groupe

Defining a group

liaison d'un jeu d'identificateurs d'objets symboliques numérotés implicitement à une entité de groupe vide

Deleting a group

suppression de l'identité de groupe du jeu d'identités légales, en désallouant le descripteur de groupe associé

Disrupting a Functional Unit or procedure

fin brutale (non ordonnée) de la Functional Unit ou de la procédure

Dynamic Association

association entre un INITIATOR UE et un RESPONDER UE, dont la création et la fin sont à la discrétion du INITIATOR UE

Function Group

ensemble nommé de Functional Units de fonctionnalités liées

Functional Unit invocation

instance spécifique de l'utilisation de la Functional Unit

Functional Unit type

ensemble nommé de Functional Units de mécanismes d'actions liées

Functional Unit, ou User Element Functional Unit

succession bien définie de primitives de service TASE.1 nommées sur les TAPI de deux systèmes TASE.1 communicants, constituant une capacité fonctionnelle de coopération unique d'un User Element INITIATOR et de son User Element RESPONDER homologue⁴⁾

Group

jeu numéroté d'objets données nommés et numérotés de façon implicite d'un système TASE.1

Incarnation

jeu cohérent de valeurs de données pour un groupe ou sous-groupe, tous échantillonnés à un moment donné

INITIATOR address

identification unique de l'INITIATOR UE

INITIATOR site

ensemble de systèmes INITIATOR partageant un Configuration Set commun. Equivalent au système INITIATOR, si un tel partage n'existe pas

INITIATOR system

ensemble comprenant tous les INITIATOR UE d'un système TASE.1 donné ainsi que l'environnement de traitement des données local auquel appartient l'ensemble

3) Numéro de groupe.

4) Il s'agit ici de la définition de base. Cependant, il est permis à une FU d'agir exclusivement par le biais d'autres FU, sans avoir de séquence de primitive de service TASE.1 spécifique associée. Pour plus d'informations sur les types de FU, voir 4.2.

Creating a group

making a new group identity³⁾ legal, allocating a new group descriptor

Defining a group

attaching a set of implicitly numbered symbolic object identifiers to an empty group identity

Deleting a group

removing the group identity from the set of legal identities, de-allocating the associated group descriptor

Disrupting a Functional Unit or procedure

abruptly (non-orderly) terminating that Functional Unit or procedure

Dynamic Association

an association between an INITIATOR UE and a RESPONDER UE, which may be created and terminated at the discretion of the INITIATOR UE

Function Group

a named collection of Functional Units of related functionalities

Functional Unit invocation

a specific instance of use of the Functional Unit

Functional Unit type

a named collection of Functional Units of related action mechanisms

Functional Unit, or User Element Functional Unit

a named well-defined succession of TASE.1 service primitives at the TAPIs of two communicating TASE.1 systems, constituting a single cooperative functional capability of a INITIATOR User Element and its peer RESPONDER User Element⁴⁾

Group

a numbered set of named, and implicitly numbered, data objects in a TASE.1 system

Incarnation

a consistent set of data values for a group or subgroup, all sampled at a given point in time

INITIATOR address

the unique identification of INITIATOR UE

INITIATOR site

the collection of INITIATOR systems sharing a common Configuration Set. Equivalent to INITIATOR system, if no such sharing

INITIATOR system

the collection of all INITIATOR UEs in a given TASE.1 system, together with the local data processing environment of which the collection is part

³⁾ Group number.

⁴⁾ This is the basic definition. However, an FU may act exclusively through other FUs, having no specific TASE.1 service primitive sequence associated with it. See 4.2.

INITIATOR User Element, ou INITIATOR UE

User Element contrôlant les associations, les groupes et les transferts de données via le TASE.1

Managing a group

création, modification ou suppression d'un groupe

Permanent Association

association entre un INITIATOR UE et un RESPONDER UE, à maintenir en permanence

Primary FU

types élémentaires de FUs; elles se caractérisent par des séquences primitives de service TASE.1 individuelles bien définies et sont toujours initiées par un INITIATOR UE

Procedure

séquence d'actions à effectuer sur un INITIATOR UE de TASE.1 et/ou sur son RESPONDER UE homologue

Redefining a group

modification du jeu d'identificateurs d'objets existant d'un groupe défini

RESPONDER system

ensemble comprenant tous les RESPONDER UE d'un système TASE.1 donné ainsi que l'environnement de traitement des données local auquel appartient l'ensemble

RESPONDER site

ensemble de systèmes RESPONDER partageant un Configuration Set commun. En l'absence d'un tel partage, le site RESPONDER correspond au système RESPONDER

RESPONDER User Element, ou RESPONDER U

UE de communications homologue d'un INITIATOR UE

Secondary FU

les Secondary FU ont des séquences de primitives de service TASE 1 individuelles bien définies, mais sont toujours impliquées par un RESPONDER UE, comme résultat d'une décision locale dans ce User Element

Subgroup

plage contiguë d'objets dans une définition de groupe

TASE.1

Telecontrol Application Service Element no. 1. Voir article 4

TASE.1 partner

site TASE.1 avec lequel il est permis à un INITIATOR UE ou à un RESPONDER UE de communiquer par le biais de TASE.1

TASE.1 provider

composant du logiciel qui implémente le protocole TASE.1 dans un environnement donné

INITIATOR User Element, or INITIATOR UE

a User Element controlling associations, groups and data transfer, via the TASE.1

Managing a group

creating, changing or deleting a group

Permanent Association

an association between an INITIATOR UE and a RESPONDER UE, which is to be maintained at all times

Primary FU

primary FUs are the basic kind of FUs; those that are characterized by their individual well defined sequence of TASE.1 Service primitives, and always being invoked by an INITIATOR UE

Procedure

sequence of prescribed actions in a TASE.1 INITIATOR UE and/or its peer RESPONDER UE

Redefining a group

modifying the existing set of object identifiers in a defined group

RESPONDER system

the collection of all RESPONDER UEs in a given TASE.1 system, together with the local data processing environment of which the collection is part

RESPONDER site

the collection of RESPONDER systems sharing a common Configuration Set. Equivalent to RESPONDER system, if no such sharing

RESPONDER User Element, or RESPONDER UE

the peer communications UE of an INITIATOR UE

Secondary FU

secondary FUs have individual well-defined TASE.1 Service primitive sequences, but are always involved by a RESPONDER UE, as a result of local decision in that User Element

Subgroup

a contiguous range of objects within a group definition.

TASE.1

Telecontrol Application Service Element No. 1: see clause 4

TASE.1 partner

a TASE.1 site with which a given INITIATOR UE or RESPONDER UE may communicate via the TASE.1

TASE.1 provider

the software component that implements the TASE.1 protocol in a given environment

TASE.1 system

jeu de User Elements ou User Element unique, qui utilise le fournisseur de TASE.1 adressé par la partie basse d'une adresse TASE.1 donnée⁵⁾. Le reste de l'environnement de traitement des données locales auquel appartient (appartiennent) le(les) User Element(s) est considéré comme faisant partie du même système TASE.1.

Transaction, ou TASE.1 transaction

instance spécifique d'utilisation d'un service TASE.1 élémentaire

3.2 Abréviations

ADFU:	FU d'association dynamique (Dynamic Association FU)
AE:	entité application (Application Entity)
AP:	processus d'application (Application Process)
APFU:	FU d'association permanente (Permanent Association FU)
ASE:	élément du service d'application (Application Service Element)
ATFU:	FU d'association d'essai (Test Association FU)
Conf:	confirmation
CS:	jeu de configuration (Configuration Set)
CS(I):	Copie CS sur le site INITIATOR
CS(R):	Copie CS sur le site RESPONDER
DPRFU:	FU de transfert de données requises périodiquement (Periodically Requested Data Transfer FU)
DRFU:	FU de transfert des données requises (Requested Data Transfer FU)
DSFU:	FU de transfert des données de contrôle de supervision (Supervisory Control Data Transfer FU)
DUFU:	FU de transfert des données non sollicitées (Unsolicited Data Transfer FU)
DUPFU:	FU de transfert des données périodiques non sollicitées (Unsolicited Periodic Data Transfer FU)
DUSFU:	FU de transfert des données spontanées non sollicitées (Unsolicited Spontaneous Data Transfer FU)
FU:	unité fonctionnelle (de l'élément utilisateur de TASE.1) ((TASE.1 User Element) Functional Unit)
GCFU:	FU de configuration de groupe (Group Configuration FU)
GDFU:	FU de définition de groupe (Group Definition FU)
GMFU:	FU de gestion de groupe (Group Management FU)
GRFU:	FU d'affichage de groupe (Group Readout FU)
Ind:	indication
PDU:	unité de données du protocole (Protocol Data Unit)
RAFU:	redémarrage et réactivation de la FU (Restart Reactivate FU)

⁵⁾ Il est permis à plusieurs adresses TASE.1 de tous niveaux d'adresser un système TASE.1. Par exemple, plusieurs DTE différents (qui correspondent à une forme d'adresses TASE.1 de bas niveau) adresseront le même système TASE.1 si:

- Tous les numéros DTE sont connectés aux lignes X.25 associées au dispositif sur lequel le système TASE.1 est résidant, et:
- Tous les numéros DTE contiennent la sous-adresse unique définie pour TASE.1.

TASE.1 system

the set of User Elements, or single User Element, that utilize the TASE.1 provider that is addressed by the low-level part of a given TASE.1 address⁵⁾. The rest of the local data processing environment of which the User Elements (or User Element), are (is) part, is also considered to belong to the same TASE.1 system

Transaction, or TASE.1 transaction

a specific instance of use of an elementary TASE.1 service

3.2 Abbreviations

ADFU:	Dynamic Association FU
AE:	Application Entity
AP:	Application Process
APFU:	Permanent Association FU
ASE:	Application Service Element
ATFU:	Test Association FU
Conf:	confirm
CS:	Configuration Set
CS(I):	The CS copy at the INITIATOR site
CS(R):	The CS copy at the RESPONDER site
DPRFU:	Periodically Requested Data Transfer FU
DRFU:	Requested Data Transfer FU
DSFU:	Supervisory Control Data Transfer FU
DUFU:	Unsolicited Data Transfer FU
DUPFU:	Unsolicited Periodic Data Transfer FU
DUSFU:	Unsolicited Spontaneous Data Transfer FU
FU:	(TASE.1 User Element) Functional Unit
GCFU:	Group Configuration FU
GDFU:	Group Definition FU
GMFU:	Group Management FU
GRFU:	Group Readout FU
Ind:	indication
PDU:	Protocol Data Unit
RAFU:	Restart Reactivate FU

⁵⁾ A TASE.1 system may be addressed by more than one low-level TASE.1 address. For example, a number of different DTE numbers (which is one form of low-level TASE.1 addresses) will address the same TASE.1 system, if:

- All DTE numbers connect to X.25 lines attached to the device in which the TASE.1 system resides, and;
- All DTE numbers contain the single subaddress that is defined for TASE.1.

Req:	requête (request)
Resp:	réponse (response)
RRFU:	FU de redémarrage et reconfiguration (Restart Reconfigure FU)
TAPI:	interface programmatique des applications de TASE.1 (TASE.1 Application Programming Interface)
TASE.1:	élément de service d'application de téléconduite 1 (Telecontrol Application Service Element n° 1)
UE:	élément utilisateur (User Element)
UTC:	temps universel coordonné (heure de Greenwich) (Coordinated Universal Time (Greenwich Mean Time))

4 Élément utilisateur de TASE.1

4.1 Modèle conceptuel

Pour les besoins du présent rapport technique, toute entité TASE.1 (qui comprend les fonctionnalités de communication de TASE.1 du processus d'application (AP) auquel il appartient), se présente selon le modèle ci-dessous:

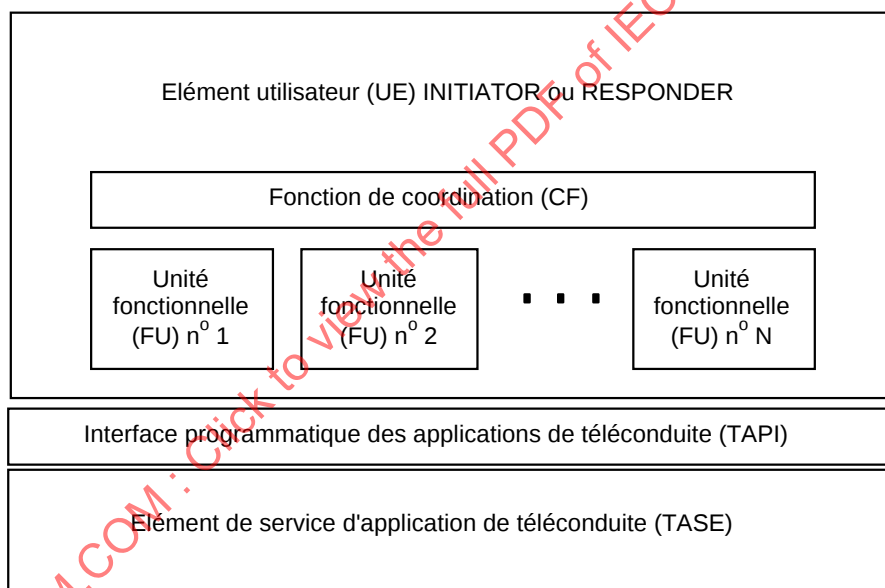


Figure 1 – Modèle conceptuel d'une entité TASE.1

Le TASE.1 UE est défini comme la partie de l'entité d'application (AE) de TASE.1 AE qui n'appartient pas à TASE.1/TAPI. Il est soit de type INITIATOR (dans un processus d'application INITIATOR), soit de type RESPONDER (dans un processus d'application RESPONDER).

Les séquences de primitives de service TASE.1 définies par le rapport technique sont représentées par les unités fonctionnelles (FU) indiquées, lesquelles sont coordonnées par une fonction de coordination (CF) unique dans chaque élément utilisateur (UE)⁶⁾.

⁶⁾ Les FU de type composite (qui n'ont pas de séquence de primitives explicite associée) sont à considérer comme associées à la séquence de primitives composites, laquelle est composée de séquences de primitives des FU «composantes» invoquées par la FU.

Req:	request
Resp:	response
RRFU:	Restart Reconfigure FU
TAPI:	TASE.1 Application Programming Interface
TASE.1:	Telecontrol Application Service Element No. 1
UE:	User Element
UTC:	Coordinated Universal Time (Greenwich Mean Time).

4 The TASE.1 User Element

4.1 Conceptual model

For the purposes of this technical report, any TASE.1 Entity (which encompasses the TASE.1 communicational functionality of the Application Process (AP) of which it is part), is modelled as shown in the figure below:

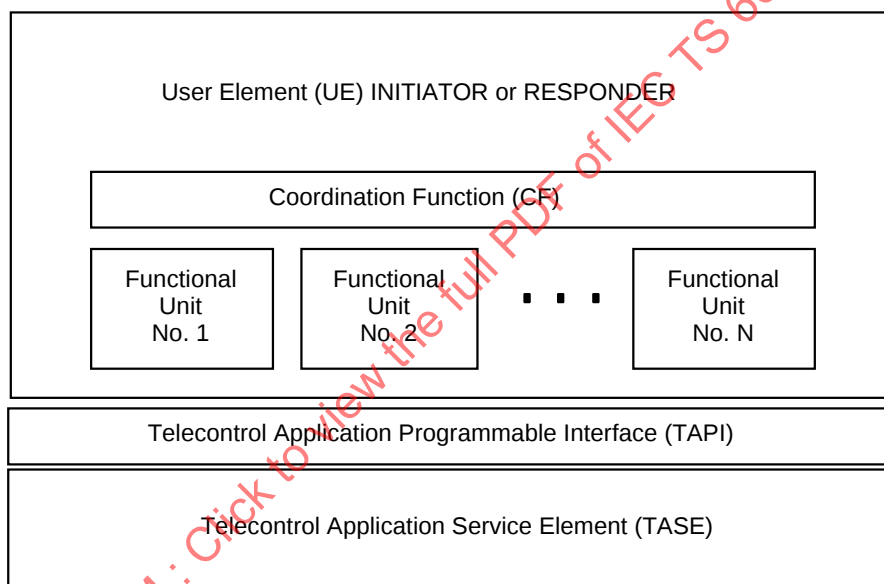


Figure 1 – TASE.1 entity conceptual model

The TASE.1 UE is defined as that part of the TASE.1 AE that is not part of the TASE.1/TAPI. It may be of either the INITIATOR type (in an INITIATOR AP) or of the RESPONDER type (in a RESPONDER AP).

The TASE.1 service primitive sequences defined by this technical report are represented by the indicated Functional Units (FUs), which are coordinated by a single Coordinating Function in each UE⁶⁾.

⁶⁾ FUs of type composite (that have no explicit primitive sequence associated with them) may be looked upon as being associated with the composite primitive sequence consisting of the primitive sequences of the "component" FUs that the FU invokes.

Il convient de souligner que ce modèle est un modèle purement conceptuel. Il ne correspond en aucun cas à une spécification d'implémentation relative à l'existence d'une FU ou d'une entité CF réelles d'une UE. L'ordre hiérarchique utilisé pour les FU et la CF n'est pas non plus obligatoire.

Le présent rapport technique spécifie un certain nombre d'invocations et de ruptures de FU effectuées par d'autres FU. Ce mécanisme est utilisé uniquement pour faciliter la description de l'ordre temporel et des éventuelles fins brutales des différentes séquences de primitives de service TASE.1. Il ne doit pas être considéré comme nécessaire du point de vue de l'architecture⁷⁾.

La CF représente toutes les interactions qui existent entre des FU et des fonctions d'ordre supérieur, ainsi qu'entre les invocations FU elles-mêmes. Les descriptions détaillées des différentes FU (voir ci-dessous) exposent les fonctionnalités des CF requises, dans les paragraphes traitant des relations avec les autres FU et des règles de coordination.

En résumé:

Les FU correspondent à des *objets utilisés pour spécifier des séquences de primitives TASE.1*, et les CF correspondent à un *objet de description des règles de coordination entre FU*.

La CF ne fait pas en elle-même l'objet d'une description détaillée dans ce rapport technique, ce qui permet d'utiliser tout type d'implémentation, si les règles définies sont respectées.

Une description très détaillée des différentes FU est présentée ci-dessous.

4.2 Unités fonctionnelles (FU)

4.2.1 Généralités

Les FU sont généralement composées de deux parties: une partie INITIATOR et une partie RESPONDER. Tout système de type INITIATOR supportant une FU donnée doit implémenter la partie INITIATOR de cette FU. De même, tout système de type RESPONDER doit implémenter la partie RESPONDER de toute FU qu'il supporte.

Les parties INITIATOR et RESPONDER d'une FU sont indépendamment invoquées et terminées, en fonction d'événements généralement différents pour l'INITIATOR et le RESPONDER. Cependant, toute invocation de FU dans laquelle l'une des deux parties ne parvient pas à la seconde finit par se terminer, ceci en raison de la mise en place d'une action de traitement des erreurs locales.

Les invocations de FU multiples et concourantes sont généralement légales. Le présent rapport technique impose des limites à certaines FU. Ces limites sont définies dans la description des différentes FU⁸⁾.

Les erreurs TAPI locales de type «*call not accepted due to flow control*» doivent être traitées de la manière suivante:

⁷⁾ Par exemple, il est permis de faire contrôler directement toutes les invocations de FU par la CF. Il est également permis d'invoquer une FU à partir d'une autre FU, en effectuant une requête à partir de la FU «d'invocation» vers la CF.

⁸⁾ Il est à noter que TASE.1 impose ses propres règles de mise en séquence de primitives, qui s'ajoutent à celles décrites dans le présent rapport technique. Voir l'article 1.

It should be noted that this is a purely conceptual model. It is not to be interpreted as an implementation specification with respect to the existence of any actual FU or Coordinating Function entity in a UE. Neither is the indicated hierarchical ordering of FUs and the Coordinating Function mandatory.

This technical report specifies a number of FU invocations and disruptions by other FUs. This mechanism is a purely notational convenience for describing the temporal ordering and possible abrupt termination of the various TASE.1 service primitive sequences, and shall not be interpreted as an architectural requirement⁷⁾.

The Coordinating Function represents all interactions between FUs and higher-order functions, as well as between the FU invocations themselves. In the detailed descriptions of the individual FUs below, the required Coordinating Function functionalities are stated in the subclauses "Relation to other FUs" and "Coordination rules".

To summarize:

FUs function as *objects for TASE.1 primitive sequence specifications*, and the Coordinating Function functions as an *object for describing the coordination rules between FUs*.

The Coordinating Function itself is not described in any more detail by this technical report, allowing for any manner of implementation as long as the stated rules are adhered to.

The individual FUs are described in considerable detail below.

4.2 Functional units (FUs)

4.2.1 General

FUs generally consist of an INITIATOR part and a RESPONDER part. Any INITIATOR type system supporting a given FU shall implement the INITIATOR part of that Functional Unit. Correspondingly, any RESPONDER type system shall implement the RESPONDER part of any FU it supports.

The INITIATOR and the RESPONDER parts of any FU are invoked and terminated independently, upon events that generally differ in the INITIATOR and RESPONDER. However, any FU invocation in which the one part does not eventually succeed in invoking the other, will be terminated in due course, because of local error handling action.

Concurrent and multiple Functional Unit invocations are generally legal. Restrictions that this technical report imposes on a given Functional Unit in this respect are listed in the description of that Functional Unit, below⁸⁾.

Local TAPI errors of the type *"call not accepted due to flow control"* shall be treated in the following manner:

⁷⁾ For example, all FU invocations may be directly controlled by the Coordinating Function, and an FU invocation by another FU may be effected by request from the "invoking" FU to the Coordinating Function.

⁸⁾ Note, however, that the TASE.1 imposes its own set of primitive sequencing rules, in addition to those mentioned here. See clause 1.

soit:

- effectuer l'action spécifiée dans la partie «congestion error» de la description de FU ci-dessous;

soit:

- suspendre la FU pendant un temps déterminé localement et relancer. Effectuer le cycle suspendre/relancer jusqu'à ce que la FU réussisse, qu'une autre erreur soit générée, ou que le nombre d'essais défini localement soit atteint;
- si l'opération réussit, suivre les procédures habituelles;
- si l'erreur «*call not accepted due to flow control*» persiste: effectuer l'action spécifiée dans la partie «congestion error» de la description de FU ci-dessous;
- si une autre erreur est générée: suivre les instructions données dans la description de FU ci-dessous.

4.2.2 Vue d'ensemble

Le concept de FU a été présenté dans les paragraphes précédents. Ce paragraphe énumère les différentes FU définies par le présent rapport technique, accompagnées d'une brève description de leur domaine d'application. Afin de simplifier la présentation, les FU sont regroupées en Function Groups, chacun de ces groupes contenant les FU d'une «famille» de fonctions précise.

Chaque FU est classée selon l'un des trois types de Functional Unit, qui reflètent les différents modes d'action: FU primary, secondary et composite.

Les Primary FU correspondent aux FU de base. Elles se caractérisent par leur séquence de primitives de service TASE.1 individuelle bien définie et sont toujours invoquées par un INITIATOR UE.

Exemple: la *Unsolicited Data Transfer FU*.

Les Secondary FU possèdent des séquences de primitives de service TASE.1 individuelles bien définies, mais sont toujours invoquées par un RESPONDER UE, en raison d'une décision locale de ce UE.

Exemple: la *Unsolicited Periodic Data Transfer FU*.

Les Composite FU agissent par le biais de l'invocation d'autres FU par un INITIATOR UE uniquement. Elles ne possèdent pas de séquence de primitives de service TASE.1 spécifique associée. Elles ne disposent pas non plus d'une partie RESPONDER spécifique.

Exemple: la *Restart Reconfigure FU*.

Une FU est identifiée par son abréviation, dont la première lettre identifie son Function Group:

A: Gestion d'association

G: Identification de données

D: Transfert de données

R: Redémarrage

FUNCTION GROUP GESTION D'ASSOCIATION: A

Permanent Association FU. Abréviation: APFU. Type: Primary. Création d'une association unique entre un INITIATOR UE et un RESPONDER UE, et recreation de cette association à chaque rupture. Les associations traitées par cette FU sont dites permanent.

either:

- perform the action specified under the heading "congestion error", in the FU description below;

or:

- suspend the FU for a locally determined amount of time, then retry. Perform the suspend/retry cycle until success, other error or a locally determined maximum number of retrials has been performed;
- if success: proceed as usual;
- if the "*call not accepted due to flow control*" has occurred persistently: perform the action specified under the heading "congestion error", in the FU description below;
- if other error: handle as specified in the FU description, below.

4.2.2 Overview

The FU concept is presented in the preceding subclauses. In this subclause each FU defined by this technical report is listed, together with a short description of its scope. For ease of presentation, the FUs are grouped into Function Groups, each of which encompasses the FUs of a specific "family" of functions.

Each FU is assigned to one of three defined Functional Unit types, reflecting different action modes: primary, secondary and composite FUs.

Primary FUs are the basic kind of FUs; those that are characterized by their individual well-defined sequence of TASE.1 service primitives, and always being invoked by an INITIATOR UE.

Example: the *Unsolicited Data Transfer FU*.

Secondary FUs have individual well-defined TASE.1 service primitive sequences, but are always invoked by a RESPONDER UE as a result of local decision in that UE.

Example: the *Unsolicited Periodic Data Transfer FU*.

Composite FUs act via invocation of other FUs by an INITIATOR UE only. They have no associated specific TASE.1 service primitive sequence. Neither do they have any specific RESPONDER part.

Example: the *Restart Reconfigure FU*.

An FU is identified by its individual acronym, the first letter of which identifies its Function Group:

A: Association Management

G: Data Identification

D: Data Transfer

R: Restart

THE ASSOCIATION MANAGEMENT FUNCTION GROUP: A

The *Permanent Association FU*. Acronym: APFU. Type: Primary. Creation of a single association between an INITIATOR UE and a RESPONDER UE, and re-creating it whenever disrupted. Associations handled by this FU are referred to as permanent.

Dynamic Association FU. Abréviation: ADFU. Type: Primary. Création d'une association unique entre un INITIATOR UE et un RESPONDER UE, et enregistrement de sa rupture éventuelle ou de sa fin ordonnée. Les associations créées par cette FU sont dites dynamic.

Test Association FU. Abréviation: ATFU. Type: Primary. Vérification de l'accessibilité d'un RESPONDER UE à partir d'un INITIATOR UE.

FUNCTION GROUP IDENTIFICATION DE DONNÉES: G

Group Configuration FU. Abréviation: GCFU. Type: Composite. Création et définition d'un groupe dans un CS(R), à la demande d'un INITIATOR UE.

Group Management FU. Abréviation: GMFU. Type: Primary. Création, modification ou suppression d'un groupe dans un CS(R), à la demande d'un INITIATOR UE.

Group Definition FU. Abréviation: GDFU. Type: Primary. Définition d'un groupe dans un CS(R), à la demande d'un INITIATOR UE.

Group Readout FU. Abréviation: GRFU. Type: Primary. Transfert de l'affichage d'une configuration de groupe d'un CS(R) vers un INITIATOR UE, à la demande de ce dernier.

FUNCTION GROUP TRANSFERT DE DONNÉES: D

Requested Data Transfer FU. Abréviation: DRFU. Type: Primary. Transfert de données associées auparavant à un groupe unique, d'un RESPONDER UE vers un INITIATOR UE, à la demande de ce dernier.

Periodically Requested Data Transfer FU. Abréviation: DPRFU. Type: Composite. Transfert de données associées à un groupe unique, d'un RESPONDER UE vers un INITIATOR UE, sur demandes périodiques de ce dernier.

Unsolicited Data Transfer FU. Abréviation: DUFU. Type: Primary. Autorisation accordée au RESPONDER UE de transférer des données d'un groupe spécifié, périodiquement ou spontanément.

Unsolicited Periodic Data Transfer FU. Abréviation: DUPFU. Type: Secondary. Transfert périodique de données associées à un ou plusieurs groupes quels qu'ils soient, d'un RESPONDER UE vers un INITIATOR UE, qui a autorisé le transfert périodique de données pour le groupe en question. L'autorisation et le retrait de cette autorisation font partie de la *Unsolicited Data Transfer FU*.

Unsolicited Spontaneous Data Transfer FU. Abréviation: DUSFU. Type: Secondary. Transfert de données associées à un ou plusieurs groupes quels qu'ils soient, d'un RESPONDER UE vers un INITIATOR UE, un nombre de fois non défini, sur décision du RESPONDER UE, et après autorisation par l'INITIATOR UE. L'autorisation et le retrait de cette autorisation font partie de la *Unsolicited Data Transfer FU*.

Supervisory Control Data Transfer FU. Abréviation: DSFU. Type: Primary. Transfert de commandes fonctionnelles ou de valeurs de consigne d'un INITIATOR UE vers un RESPONDER UE.

FUNCTION GROUP REDÉMARRAGE: R

Restart Reconfigure FU. Abréviation: RRFU. Type: Composite. Fourniture, par un INITIATOR UE d'un CS(R) complet à un RESPONDER UE, sur requête de ce dernier et si l'INITIATOR l'estime approprié. Le mécanisme de requête fait partie de la *Permanent Association FU* et de la *Dynamic Association FU*, mais pas de la *Restart Reconfigure FU* elle-même.

The *Dynamic Association FU*. Acronym: ADFU. Type: Primary. Creation of a single association between an INITIATOR UE and a RESPONDER UE, registering its eventual disruption or eventually terminating it in an orderly manner. Associations created by this FU are referred to as dynamic.

The *Test Association FU*. Acronym: ATFU. Type: Primary. Checking the reachability of a RESPONDER UE from an INITIATOR UE.

THE DATA IDENTIFICATION FUNCTION GROUP: G

The *Group Configuration FU*. Acronym: GCFU. Type: Composite. Creating and defining a group in a CS(R) on request by an INITIATOR UE.

The *Group Management FU*. Acronym: GMFU. Type: Primary. Creating, changing or deleting a group in a CS(R) on request by an INITIATOR UE.

The *Group Definition FU*. Acronym: GDFU. Type: Primary. Defining a group in a CS(R) on request by an INITIATOR UE.

The *Group Readout FU*. Acronym: GRFU. Type: Primary. Transferring a readout of a group configuration from a CS(R) to an INITIATOR UE, on request by the latter.

THE DATA TRANSFER FUNCTION GROUP: D

The *Requested Data Transfer FU*. Acronym: DRFU. Type: Primary. Transferring data associated with a single group once, from a RESPONDER UE to an INITIATOR UE, on request by the latter.

The *Periodically Requested Data Transfer FU*. Acronym: DPRFU. Type: Composite. Transferring data associated with a single group, from a RESPONDER UE to an INITIATOR UE, on periodical requests by the latter.

The *Unsolicited Data Transfer FU*. Acronym: DUFU. Type: Primary. Granting permission to the RESPONDER UE to transfer data from one specified group, either periodically or spontaneously.

The *Unsolicited Periodic Data Transfer FU*. Acronym: DUPFU. Type: Secondary. Transferring data associated with any group(s) periodically, from a RESPONDER UE to an INITIATOR UE, which has granted permission for periodic data transfer concerning the group in question. Permission granting and withdrawal is part of the *Unsolicited Data Transfer FU*.

The *Unsolicited Spontaneous Data Transfer FU*. Acronym: DUSFU. Type: Secondary. Transferring data associated with any group(s), from a RESPONDER UE to an INITIATOR UE, any number of times at the RESPONDER UE's decision, after permission has been granted by the INITIATOR UE. Permission granting and withdrawal is part of the *Unsolicited Data Transfer FU*.

The *Supervisory Control Data Transfer FU*. Acronym: DSFU. Type: Primary. Transferring functional commands or setpoints from an INITIATOR UE to a RESPONDER UE.

THE RESTART FUNCTION GROUP: R

The *Restart Reconfigure FU*. Acronym: RRFU. Type: Composite. Furnishing a RESPONDER UE with a complete CS(R), by an INITIATOR UE, as seen appropriate by the latter after being requested by the former. The request mechanism is part of the *Permanent Association FU* and the *Dynamic Association FU*, and not of the *Restart Reconfigure FU* itself.

Restart Reactivate FU. Abréviation: RAFU. Type: Composite. Autorisation accordée à un RESPONDER UE de transmettre des données, qu'elles soient périodiques ou spontanées, par un INITIATOR UE, pour tous les groupes que ce dernier juge appropriés, et sur requête du RESPONDER UE. Le mécanisme de requête fait partie de la *Permanent Association FU* et de la *Dynamic Association FU*, mais pas de la *Restart Reactivate FU* elle-même.

4.3 Structure commune des descriptions de FU

Les descriptions des différentes FU suivent un modèle commun:

Fonction:

Brève description de la fonction normale

Règles de coordination:

Liste des règles à observer par la Coordinating Function, qui gère les invocations des Functional Units:

Utilisation d'une association:

Règles à respecter pour la ou les associations utilisées par la Functional Unit.

Relation avec les autres FU:

Invoking FUs:

Liste d'autres Functional Units autorisées à invoquer la Functional Unit.

Invoked FUs:

Liste d'autres Functional Units invocables par la Functional Unit.

Disrupting FUs:

Liste des Functional Units qui risquent d'interrompre (terminer brutalement) la Functional Unit.

Disrupted FUs:

Liste des Functional Units qui risquent d'être interrompues (terminées brutalement) par la Functional Unit.

Invocation:

Conditions préalables:

Liste des conditions à remplir avant d'invoquer la Functional Unit.

Restrictions:

Liste des combinaisons d'invocations illégales impliquant la Functional Unit.

Invoking events:

Description des mécanismes d'invocation de la Functional Unit.

Fin:

Orderly termination:

Description des mécanismes de fin ordonnée de la Functional Unit.

Disruption:

Description des mécanismes de fin brutale de la Functional Unit.

Procédures:

Actions à effectuer dans la Functional Unit.

Primitives de service TASE.1:

Liste des services TASE.1 élémentaires utilisés par la Functional Unit, avec une description du fonctionnement sans erreur:

The *Restart Reactivate FU*. Acronym: RAFU. Type: Composite. Granting, to a RESPONDER UE, permission to transmit data, both periodic and spontaneous, by an INITIATOR UE for all groups that the latter judges appropriate, after being requested by the former. The request mechanism is part of the *Permanent Association FU* and the *Dynamic Association FU*, and not of the *Restart Reactivate FU* itself.

4.3 Common template for the descriptions of FUs

The individual Functional Unit descriptions conform to a common template:

Function:

A short description of the normal function

Coordination rules:

A list of rules to be observed by the Coordinating Function governing Functional

Unit invocations:

Association usage:

Requirements as to the association(s) that are used by the Functional Unit.

Relation to other FUs:

Invoking FUs:

List of other Functional Units that may invoke the Functional Unit.

Invoked FUs:

List of other Functional Units that the Functional Unit may invoke.

Disrupting FUs:

List of Functional Units that may disrupt (abruptly terminate) the Functional Unit

Disrupted FUs:

List of Functional Units that may be disrupted (abruptly terminated) by the Functional Unit.

Invocation:

Prerequisites:

List of conditions to be met before invoking the Functional Unit.

Restrictions:

List of illegal invocation combinations involving the Functional Unit.

Invoking events:

Description of the Functional Unit invocation mechanisms.

Termination:

Orderly termination:

Description of the mechanisms for orderly termination of the Functional Unit.

Disruption:

Description of the mechanisms for abrupt termination of the Functional Unit.

Procedures:

Prescribed actions within the Functional Unit.

TASE.1 service primitives:

List of elementary TASE.1 services used by the Functional Unit, with description of error-free operation:

Sequence:

Séquence normale des primitives associées à la Functional Unit.

Parameter values:

Valeurs des paramètres des primitives émises vers TASE.1, dans le cadre d'une séquence de primitives normale.

Traitement des erreurs:

Jeu de règles définissant les conditions de traitement des erreurs intervenant dans la Functional Unit:

FU disruption:

Traitement d'une fin brutale de la Functional Unit.

Illegal invocation attempt:

Traitement des tentatives d'invocation de la Functional Unit, qui, selon les règles définies dans la section Restrictions (ci-dessus), ne sont pas légales.

Incoming TASE.1 service primitive out of context:

Traitement des primitives reçues de TASE.1, avec des valeurs de paramètres légales, mais intervenant à un moment inopportun.

Timing errors:

Traitement d'une réponse manquante ou excessivement tardive de la part du User Element homologue d'un INITIATOR ou d'un RESPONDER User Element.

Congestion error:

Traitement des situations dans lesquelles la TAPI n'est pas en mesure d'accepter une autre primitive sortante en raison de sa charge lourde.

TASE.1 service primitive parameter errors:

Traitement des valeurs de paramètres erronées des primitives *entrantes* de TASE.1, les primitives elles-mêmes intervenant dans le cadre d'une séquence normale.

5 Function group gestion d'association

5.1 Adressage

La fonction d'une adresse TASE.1 est de spécifier une Application Entity dans un système TASE.1 cible. Les adresses TASE.1 sont divisées en deux parties: Le titre du processus d'application, nommé «adresse» dans ce rapport et l'application entity qualifier, nommé «A-suffix» dans ce rapport technique. L'adresse spécifie un processus d'application de téléconduite se situant uniquement dans le cadre de tous les processus d'application de téléconduite, alors que le A-suffix spécifie une Application Entity se situant uniquement dans le cadre de toutes les Application Entities du processus d'application de téléconduite adressées.

Sequence:

Normal sequence of primitives associated with the Functional Unit.

Parameter values:

Values of parameters of primitives *issued* to the TASE.1 in the course of normal primitive sequence.

Error handling:

Set of rules for handling error conditions within the Functional Unit:

FU disruption:

Handling of abrupt termination of the Functional Unit.

Illegal invocation attempt:

Handling of Functional Unit invocation attempts that are not legal, according to the rules stated in section Restrictions (above).

Incoming TASE.1 service primitive out of context:

Handling of primitives *received* from the TASE.1, with legal parameter values, but occurring at the wrong time.

Timing errors:

Handling of missing or excessively delayed response on the part of the peer User Element of an INITIATOR or RESPONDER User Element.

Congestion error:

Handling of situations where the TAPI is not able to accept another outgoing primitive because of heavy load.

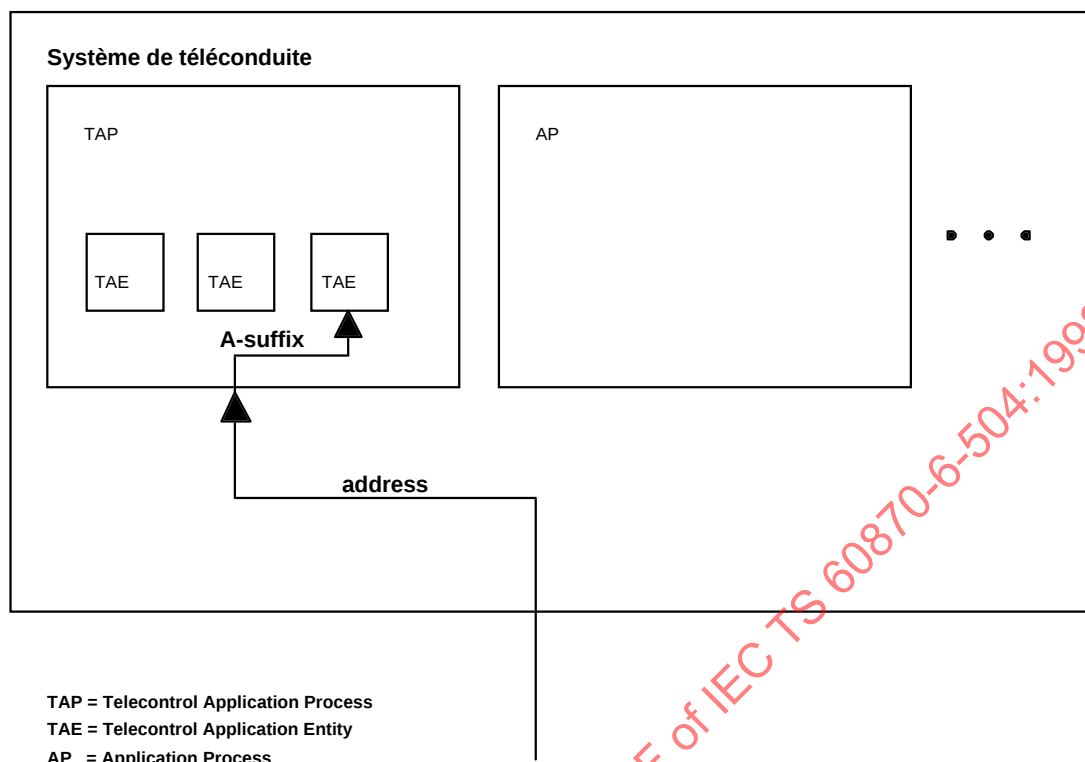
TASE.1 service primitive parameter errors:

Handling of erroneous parameter values in *incoming* primitives from the TASE.1, the primitives themselves occurring within normal sequence.

5 The Association Management Function Group

5.1 Addressing

The function of a TASE.1 address is to specify an Application Entity in a target TASE.1 system. TASE.1 addresses are composed of two parts: The application process title, hereafter called "address", and the application entity qualifier, hereafter called "A-suffix". The address specifies a telecontrol application process uniquely within the space of all telecontrol application processes, while the A-suffix specifies an Application Entity uniquely within the space of all Application Entities within the addressed telecontrol application process.



La recherche de l'adresse de présentation d'une adresse donnée se fait au niveau local. Il est permis d'utiliser le OSI Directory dans ce but.

La *Permanent Association FU* et la *Dynamic Association FU* utilisent les adresses TASE.1 et les A-suffixes codés dans les paramètres Initiator et Acceptor, afin d'établir une association (voir ci-dessous).

Tout UE conforme à ce rapport doit être capable de supporter les invocations de la *Permanent Association FU* et de la *Dynamic Association FU*, afin de pouvoir établir des associations avec un site INITIATOR ou RESPONDER par le biais de plusieurs adresses de TASE.1 par site. Chaque adresse TASE.1 doit correspondre à un système INITIATOR ou RESPONDER sur le site.

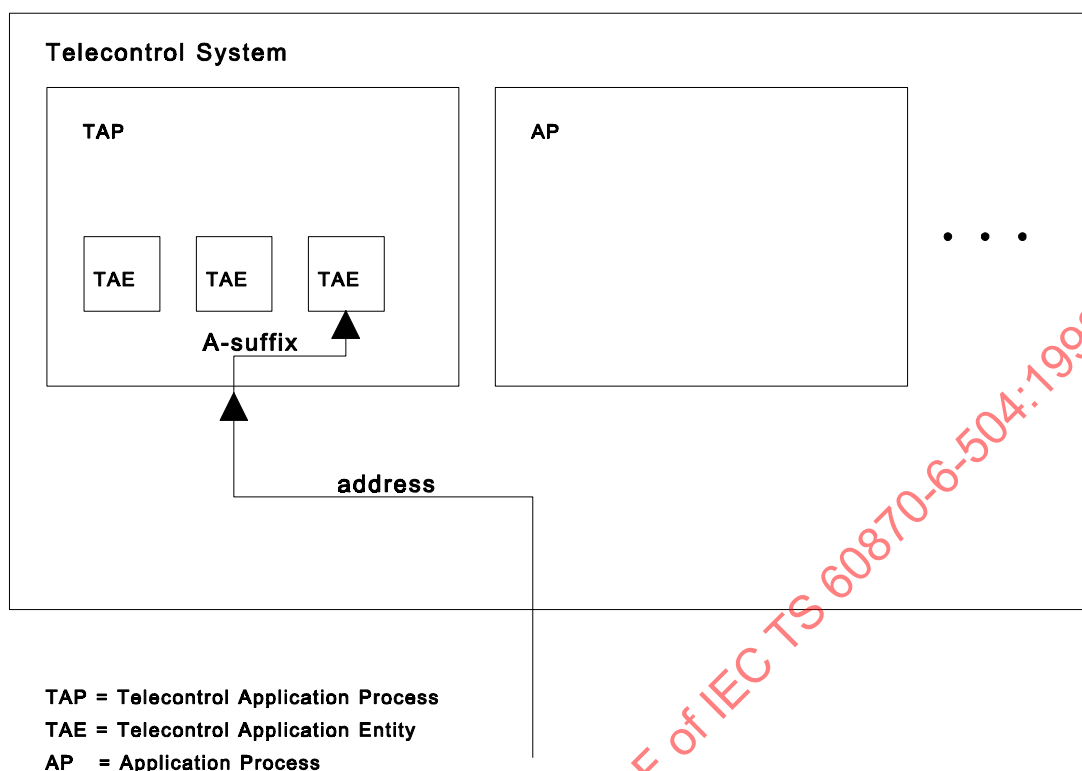
Le choix de l'adresse TASE.1 se fait au niveau de la Coordinating Function.

5.2 Partie adresse

L'adresse (titre du processus d'application) doit être transformée en une adresse de présentation avant d'appeler le service *TASEBind*. La transformation intervient au niveau local (il est cependant permis d'utiliser le OSI Directory⁹⁾).

L'adresse de présentation est composée de quatre éléments: sélecteur de présentation, sélecteur de session, sélecteur de transport et adresses réseau.

⁹⁾ L'adresse sera ensuite créée sous la dénomination «Distinguished Name», qui correspond à l'entrée au OSI Directory. Le OSI Directory répond avec l'adresse de présentation, et l'application ajoute le A-suffix (ou l'Application Entity Qualifier) avant d'appeler le *TASEBind*.



It is a local matter to find the presentation address for a given address. The OSI Directory may be used for that purpose.

The *Permanent Association FU* and the *Dynamic Association FU* utilize TASE.1 addresses and A-suffixes coded into parameters Initiator and Acceptor for association establishment; see below.

Any UE Conforming to this report shall be able to support invocations of both the *Permanent Association FU* and the *Dynamic Association FU* in order to establish associations with an INITIATOR or RESPONDER site via more than one TASE.1 address per site. Each TASE.1 address shall correspond to a component INITIATOR or RESPONDER system within the site.

TASE.1 address selection is a function at Coordinating Function level.

5.2 Address part

The address (application process title) has to be transformed into a presentation address before calling the *TASEBind*. The transformation is a local matter, however the OSI Directory may be used⁹⁾.

The presentation address consists of four components: presentation selector, session selector, transport selector, and network addresses.

⁹⁾ The address will then be built up as a so-called "Distinguished Name" which is the entry to the OSI Directory. The OSI Directory responds with the presentation address, and the application has to add the A-suffix (or Application Entity Qualifier) before calling the *TASEBind*.

5.2.1 A-suffices

Les A-suffices sont des composantes des paramètres Initiator et Acceptor des primitives *TASEBind*. Seules les valeurs suivantes sont à considérer comme norme dans le contexte de ce rapport :

Fonction	Functional Units	Initiator	Responder
Accès au CS(R) à partir de l'INITIATOR	GMFU, GDFU, GRFU	'AA'	'BA'
Transfert de données spontanées non sollicitées	DUFU, DUSFU, DRFU ¹⁰⁾	'AB'	'BB'
Transfert de données périodiques non sollicitées	DUFU, DUPFU, DRFU ¹¹⁾	'AC'	'BC'
Transfert de données requises, programmation des informations ¹²⁾	DRFU	'AD'	'BD'
Transfert de données requises, informations actuelles ou archivées	DRFU	'AE'	'BE'
Association d'essai, lorsque l'association est établie uniquement dans ce but	ATFU	'AF'	'BF'
Transfert de données de contrôle de supervision	DSFU	'AG'	'BG'
NOTE – 'AB' correspond à la représentation en ASCII de la paire de caractères AB, c'est-à-dire à la paire de valeurs décimales 65, 66.			

Les A-suffices non définis dans le tableau ci-dessus sont légaux, mais uniquement dans le cadre d'un accord bilatéral (conventions locales). De tels A-suffices «privés» ne doivent jamais commencer par les caractères 'A' ou 'B', qui sont réservés pour les A-suffices normalisés par ce rapport.

Exemple: Si un INITIATOR UE veut initialiser un transfert de données spontanées non sollicitées à partir d'un RESPONDER UE, il convient qu'il utilise le A-suffix 'AB' dans le paramètre Initiator et le A-suffix 'BB' dans le paramètre Acceptor de la primitive *TASEBindReq* qu'il émettra afin d'établir une association sur laquelle il pourra ensuite autoriser l'envoi de données non sollicitées.

5.2.2 Codage d'adresses

Tout paramètre d'adresse Initiator ou Acceptor doit être composé d'une chaîne d'octets ayant les interprétations suivantes (l'octet n° 0 indique le premier octet de la chaîne, à l'adresse la plus basse):

- Octet n° 0: Nombre binaire, exprimant le nombre d'octets de la partie correspondant à l'adresse (= N)
- Octets n° 1 à N: partie correspondant à l'adresse

¹⁰⁾ Il est permis de limiter une implémentation du DRFU uniquement à un support d'informations actuelles (et non à des informations d'archive) sur une association dans laquelle une *Unsolicited Spontaneous Data Transfer FU* est déjà lancée.

¹¹⁾ La DRFU est légale uniquement si aucune invocation de *Periodic data Transfer FU* n'est actuellement active sur l'association considérée.

¹²⁾ Les programmes (futurs et anciens) d'exploitation de centrales électriques, l'échange d'énergie, les décharges etc.

5.2.1 A-suffices

A-suffices are components of the Initiator and Acceptor parameters of the *TASEBind* primitives. Only the following values are to be considered as standard within the context of this report:

Function	Functional Units	Initiator	Responder
Accessing the CS(R) from the INITIATOR	GMFU, GDFU, GRFU	'AA'	'BA'
Unsolicited Spontaneous Data Transfer	DUFU, DUSFU, DRFU ¹⁰⁾	'AB'	'BB'
Unsolicited Periodic Data Transfer	DUFU, DUPFU, DRFU ¹¹⁾	'AC'	'BC'
Requested Data Transfer, scheduling information ¹²⁾	DRFU	'AD'	'BD'
Requested Data Transfer, pre-sent or archived information	DRFU	'AE'	'BE'
Test Association, when association is established solely for this purpose	ATFU	'AF'	'BF'
Supervisory Control Data Transfer	DSFU	'AG'	'BG'
NOTE – 'AB' means the ASCII representation of the character pair AB, i.e. the decimal value pair 65, 66.			

A-suffices not defined by the table above are legal, but only on the basis of bilateral agreement (local conventions). Such "private" A-suffices shall never begin with characters 'A' or 'B', which are reserved for A-suffices standardized by this report.

Example: If an INITIATOR UE wants to initiate unsolicited spontaneous data transfer from a RESPONDER UE, it should use the A-suffix 'AB' in the Initiator parameter and the A-suffix 'BB' in the Acceptor parameter of the *TASEBindReq* primitive, it will issue in order to establish an association over which it later will grant permission to send the unsolicited data.

5.2.2 Coding of addresses

Any Initiator or Acceptor address parameter shall be a string of octets with the following interpretation, with octet No. 0 denoting the first octet in the string, at lowest address:

- Octet No. 0: Binary number, expressing the number of octets in the address part (= N)
- Octets No. 1 through N: address part

¹⁰⁾ An implementation of the DRFU may be restricted to only support present information (no archive information) on an association where an Unsolicited Spontaneous Data Transfer FU is already running.

¹¹⁾ DRFU is legal only if no Periodic data Transfer FU invocation is currently active on the association in question.

¹²⁾ Schedules (future and historic) power plant operation, energy exchange, discharges etc.

- Octet n° N+1: Nombre binaire exprimant le nombre d'octets de la partie correspondant au A-suffix (=M ¹³)
- Octets n° N+2 à N+M+1: partie correspondant au A-suffix
(Nombre total d'octets: N+M+2.)

Ce rapport ne spécifie pas le codage des adresses de présentation. Les adresses de présentation doivent être codées conformément aux spécifications des produits de communication sous-jacents.

5.3 Marquage de redémarrage

Il est permis à un RESPONDER UE d'utiliser une restart marking dans le paramètre User-data de la primitive *TASEBindResp* pour signaler qu'il a besoin d'être réinitialisé par l'INITIATOR pour ce qui concerne l'autorisation d'envoyer des données non sollicitées et/ou périodiques. L'INITIATOR répondra à sa requête en invoquant la *Restart Reactivate FU* du système RESPONDER considéré. Seuls les codes de redémarrage suivants sont légaux dans le cadre de ce rapport¹⁴:

Valeur	Interprétation	Functional Unit
'0' (= décimal 48)	<i>No restart</i> : Réinitialisation non nécessaire	–
'1' (= décimal 49)	(Réservé à une utilisation future)	–
'2' (= décimal 50)	<i>Restart; spontaneous management lost</i> : Répétition requise pour toutes les autorisations à émettre	RAFU

La RESPONDER UE doit toujours utiliser le code de redémarrage = '2' pour signaler les pertes de gestion spontanées. Cette méthode risque d'entraîner un surdébit mais permet de rétablir la communication de façon non ambiguë.

Les codes de redémarrage sont significatifs pour l'INITIATOR, uniquement lorsque le paramètre Result = *ok* dans la primitive *TASEBindResp*.

5.4 Contrôle d'intégrité de la configuration de groupe

TASE.1 fournit un mécanisme de détection des absences de correspondance entre le CS(I) (copie du Configuration Set courant de l'INITIATOR) et le CS(R) (copie du Configuration Set courant du RESPONDER). Ce mécanisme se base sur une valeur Control Field (CF) générée sur le site RESPONDER et stockée avec le CS(R) sur le site RESPONDER et avec le CS(I) sur le site INITIATOR.

Le mécanisme de contrôle d'intégrité est divisé en deux procédures: une pour la mise à jour du Configuration Set et une pour la vérification du Configuration Set.

Procédure de mise à jour du Configuration Set:

Lorsque la *Group Management FU* ou la *Group Definition FU* est invoquée, la partie RESPONDER doit générer une valeur Control Field lorsqu'elle reçoit la primitive *GroupMgmt Ind* ou *DefineGroupInd* valide correspondante.

¹³) M est toujours égal à 2 pour les A-suffices normalisés par ce rapport.

¹⁴) Voir la description de ASN.1, dans la CEI 60870-6-502 pour le codage des User-Data.

- Octet No. N+1: Binary number, expressing the number of octets in the A-suffix part (=M)¹³⁾
- Octets No. N+2 through N+M+1: A-suffix part
(Total number of octets: N+M+2.)

This report does not specify the coding of presentation addresses. Presentation addresses shall be coded according to the specifications of the underlying communications products.

5.3 Restart Marking

A RESPONDER UE may use a restart marking in the User-data parameter of the *TASE BindResp* primitive in order to signal its need for a re-initialization by the INITIATOR, with respect to the permission to send unsolicited and/or periodic data. The INITIATOR will eventually respond to this request by invoking the *Restart Reactivate FU* for the RESPONDER system in question. Only the following restart codes are legal within the context of this report¹⁴⁾:

Value	Interpretation	Functional Unit
'0' (= decimal 48)	<i>No restart</i> : No re-initialization required	–
'1' (= decimal 49)	(Reserved for future use)	–
'2' (= decimal 50)	<i>Restart, spontaneous management lost</i> : Repetition of all permissions to send required	RAFU

The RESPONDER UE shall always use restart code = '2' in order to signal spontaneous management lost. This will introduce some overhead, but is an unambiguous way to communicate re-establishing.

The restart codes are significant for the INITIATOR, only when Result = *ok* in the *TASEBindResp* primitive.

5.4 Group Configuration integrity control

TASE.1 offers a mechanism for detecting mismatch between the CS(I) (INITIATOR's copy of the current Configuration Set) and the CS(R) (RESPONDER's copy of the current Configuration Set). This mechanism is based on a Control Field (CF) value generated at the RESPONDER site and stored with the CS(R) at the RESPONDER site and with the CS(I) at the INITIATOR site.

The integrity control mechanism is divided into a procedure for Configuration Set update, and a procedure for Configuration Set check.

Procedure for Configuration Set update:

Whenever the *Group Management FU* or the *Group Definition FU* is invoked, the RESPONDER part shall generate a Control Field value upon reception of the corresponding valid *GroupMgmtInd* or *DefineGroupInd* primitive.

¹³⁾ M is always =2 for the A-suffixes standardized by this report.

¹⁴⁾ See ASN.1 description IEC 60870-6-502 for coding of User-Data.

La valeur Control Field doit se présenter sous forme d'un tableau de 9 éléments. Les éléments doivent être des entiers de 16 bits¹⁵⁾. Les différents éléments doivent contenir les informations suivantes, l'élément n° 0 correspondant au premier élément, à l'adresse la plus basse:

- Eléments n° 0 à 6: Date et heure de mise à jour du CS(R) courante. Voir l'annexe A de la CEI 60870-6-501, description du *GroupMgmtResp*.
- Eléments n° 7 à 8: somme de contrôle sur 32 bits, générée à partir du contenu de l'ensemble du CS(R) après sa mise à jour:

L'algorithme de somme de contrôle n'est pas spécifié par ce rapport. L'utilisation effective de cette facilité de somme de contrôle n'est pas obligatoire: Il est permis de la désactiver entièrement, le RESPONDER définissant tous les octets de la somme de contrôle du Control Field à une valeur fixe, par exemple 0.

La date et l'heure de la valeur Control Field doivent être stockées avec le CS(R) mis à jour, sur le site RESPONDER. Elles remplacent l'élément Time précédent de la base de données de Configuration du groupe local¹⁶⁾.

Une fois la date et l'heure de la nouvelle valeur Control Field stockée, la primitive *GroupMgmtResp* ou *DefineGroupResp* mettant fin à l'invocation FU en cours doit être émise. Elle contient la nouvelle valeur Control Field du paramètre CF (y compris la somme de contrôle).

A la réception d'une primitive *GroupMgmtConf* ou *DefineGroupConf* valide, la partie INITIATOR de la FU doit enregistrer la valeur du paramètre CF de la primitive comme étant sa nouvelle valeur Control Field. Avant de mettre fin localement à l'invocation de la FU, l'INITIATOR doit stocker cette nouvelle valeur Control Field (y compris la somme de contrôle) avec le CS(I) mis à jour, en remplaçant la valeur Control Field précédente de la base de données Configuration du groupe local.

Procédure de vérification du Configuration Set:

Lorsque la *Permanent Association FU* ou la *Dynamic Association FU* sont invoquées, le paramètre User-data de la primitive *TASEBindResp* émise par la partie RESPONDER de la FU doit contenir une valeur Control Field. Cette valeur doit être codée conformément aux règles décrites à l'annexe A de la CEI 60870-6-501.

La valeur Control Field doit avoir été générée comme suit:

- La date et l'heure doivent être égales à la date et à l'heure stockées actuellement avec le CS(R).
- La somme de contrôle doit être calculée¹⁷⁾ à partir du CS(R) courant, en utilisant le même algorithme que celui utilisé au cours de la dernière mise à jour du Configuration Set.

A la réception d'une primitive *TASEBindConf* valide, la partie INITIATOR de la *Permanent Association FU* ou de la *Dynamic Association FU* doit décoder la valeur Control Field à partir du paramètre User-data de la primitive, en utilisant les mêmes règles de numérotation des octets que celles spécifiées pour la partie RESPONDER. La valeur Control Field décodée doit ensuite être comparée avec la valeur Control Field (y compris la somme de contrôle) qui réside actuellement avec le CS(I).

¹⁵⁾ Les implémentations qui possèdent des entiers «natifs» de plus de 16 bits doivent utiliser uniquement les 16 bits les plus faibles de chaque élément de l'entier pour stocker les données Control Field significatives.

¹⁶⁾ Il est également permis de stocker l'ensemble du Control Field (date/heure et somme de contrôle) avec le CS(R). Dans ce cas, il est possible que la valeur Control Field renvoyée avec les primitives *TASEBindResp* associées (voir la procédure de vérification du Configuration Set, plus loin dans ce rapport) contienne le contrôle de somme stockée plutôt que le résultat d'un autre calcul de somme de contrôle.

¹⁷⁾ Il est également permis de prendre la somme de contrôle dans la base de données locale (dans le cas où elle y est résidente). Voir la note de bas de page de la procédure de mise à jour du Configuration Set.

The Control Field value shall be an array of 9 elements. The elements shall be 16 bit integers¹⁵⁾. The individual elements shall contain the following information, element No. 0 denoting the first element, at lowest address:

- Elements No. 0 through 6: Time stamp for current CS(R) update: See annex of IEC 60870-6-501, description of *GroupMgmtResp*.
- Elements No. 7 through 8: 32-bit checksum, generated from the contents of the whole CS(R) after update:

The checksum algorithm is not specified by this report. The actual use of the checksum facility is not mandatory: it may be disabled entirely, the RESPONDER always setting all octets of the checksum part of the Control Field equal to a fixed value, for example 0.

The Time stamp of the Control Field value shall be stored with the updated CS(R) at the RESPONDER site, overwriting the previous Time element part in the local group Configuration database¹⁶⁾.

After the Time stamp of the new Control Field value has been successfully stored, the *GroupMgmtResp* or *DefineGroupResp* primitive terminating the current FU invocation shall be issued, containing the new Control Field value in the CF parameter (including the checksum).

Upon reception of a valid *GroupMgmtConf* or *DefineGroupConf* primitive, the INITIATOR part of the FU shall register the value in the CF parameter of the primitive as its new Control Field value. Before terminating the FU invocation locally, the INITIATOR shall store this new Control Field value (including the checksum) with the updated CS(I), overwriting the previous Control Field value in the local group Configuration database.

Procedure for Configuration Set check:

Whenever the *Permanent Association FU* or the *Dynamic Association FU* is invoked, the User-data parameter of the *TASEBindResp* primitive issued by the RESPONDER part of the FU shall contain a Control Field value. This value shall be coded according to the rules described in annex A of IEC 60870-6-501.

The Control Field value shall have been generated in the following manner:

- The time stamp shall be equal to the time stamp part currently stored with the CS(R).
- The checksum shall be computed¹⁷⁾ from the current CS(R), using the same algorithm as during last Configuration Set update.

The INITIATOR part of the *Permanent Association FU* or the *Dynamic Association FU* shall, upon reception of a valid *TASEBindConf* primitive, decode the Control Field value from the User-data parameter of the primitive, using the same octet numbering rules as those specified for the RESPONDER part. The decoded Control Field value shall then be compared to the Control Field value (including checksum) that currently resides with the CS(I).

¹⁵⁾ Implementations having "native" integers of more than 16 bits shall only use the 16 low-order bits of each integer element for storing significant Control Field data.

¹⁶⁾ Alternatively, the whole Control Field (time stamp plus checksum) may be stored with the CS(R). In this case the Control Field returned with subsequent *TASEBindResp* primitives (see procedure for Configuration Set check below) may contain the stored checksum instead of the result of another checksum computation.

¹⁷⁾ Alternatively, the checksum may be fetched from the local database if it resides there; see footnote in procedure for Configuration Set update.

Si les valeurs sont égales, aucune action particulière ne doit être entreprise. Dans le cas contraire, l'INITIATOR doit invoquer la *Restart Reconfigure FU* du partenaire en question.

A noter que la partie INITIATOR des différentes FU invoquées n'intervient jamais dans la génération de la valeur Control Field. Sa fonction (en ce qui concerne les Control Field) est uniquement d'enregistrer et de comparer les valeurs Control Field. Les règles de génération des valeurs Control Field (y compris les algorithmes de somme de contrôle) concernent donc les implémentations du RESPONDER individuelles.

5.5 Permanent Association FU

Type: Primary.

5.5.1 Fonction

Une invocation de la *Permanent Association FU* fournit la séquence fonctionnelle suivante:

- a) L'INITIATOR UE établit une association avec un RESPONDER UE.
- b) Si, au cours de l'établissement de l'association, le RESPONDER UE a demandé à l'INITIATOR UE de le réinitialiser, l'INITIATOR UE invoque la FU correspondante.
- c) Chaque fois que la *Permanent Association FU* enregistre ensuite l'association comme étant brisée (terminée brutalement), elle tente de rétablir l'association. Les tentatives de rétablissement cycliques se répètent indéfiniment, jusqu'à ce que l'association soit rétablie ou qu'une fonction d'ordre supérieur intervienne par le biais de la Coordinating Function sur l'INITIATOR UE.
- d) Si, après une tentative d'association, l'INITIATOR UE reçoit une primitive *TASEBindConf* avec le paramètre Result = *rejected permanent*, il convient, pour lui, de ne pas procéder à d'autres tentatives d'association.

5.5.2 Règles de coordination

5.5.2.1 Utilisation d'une association

L'invocation de FU crée et entretient elle-même une association. Aucune autre association n'est utilisée par cette invocation de FU.

5.5.2.2 Relation avec les autres FU

5.5.2.2.1 Invoking FUs

La *Permanent Association FU* ne doit être invoquée par aucune autre FU.

5.5.2.2.2 Invoked FUs

Il est permis à la *Permanent Association FU* d'invoquer les FU suivantes:

- *Restart Reconfigure FU*
- *Restart Reactivate FU*

5.5.2.2.3 Disrupting FUs

Il n'est permis à aucune autre FU d'interrompre la *Permanent Association FU*.

In case of equality, no special action shall be taken. In case of non-equality, the INITIATOR shall invoke the *Restart Reconfigure FU* for the partner in question.

Note that the INITIATOR part of the various FUs involved never takes any part in Control Field value generation: Its only function in this respect is to register and compare Control Field values. Consequently, the rules for generating Control Field values (including checksum algorithms) are left to the individual RESPONDER implementations.

5.5 Permanent Association FU

Type: Primary.

5.5.1 Function

A *Permanent Association FU* invocation provides the following functional sequence:

- a) The INITIATOR UE establishes an association with a RESPONDER UE.
- b) If the RESPONDER UE has requested a re-initialization function from the INITIATOR UE during association establishment, the corresponding FU is invoked by the INITIATOR UE.
- c) At any time later, whenever the *Permanent Association FU* registers the association as being broken (abruptly terminated), it will attempt to re-establish the association. Cyclic re-establishment attempts will proceed indefinitely, until success or intervention by a higher-order function via the Coordinating Function in the INITIATOR UE.
- d) If the INITIATOR UE, after an association attempt receives a *TASEBindConf* primitive with Result = *rejected permanent*, he should not proceed with further association attempts.

5.5.2 Coordination rules

5.5.2.1 Association usage

The FU invocation itself creates and maintains an association. No other association is used by that FU invocation.

5.5.2.2 Relation to other FUs

5.5.2.2.1 Invoking FUs

The *Permanent Association FU* shall not be invoked by any other FU.

5.5.2.2.2 Invoked FUs

The *Permanent Association FU* may invoke the following FUs:

- *Restart Reconfigure FU*;
- *Restart Reactivate FU*.

5.5.2.2.3 Disrupting FUs

The *Permanent Association FU* may not be disrupted by any other FU.

5.5.2.2.4 Disrupted FUs

Il est permis à la *Permanent Association FU* d'interrompre les FU suivantes, par suite d'une rupture (temporaire) d'association:

- *Test Association FU*
- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*
- *Group Readout FU*
- *Requested Data Transfer FU*
- *Periodically Requested Data Transfer FU*
- *Unsolicited Data Transfer FU*
- *Unsolicited Periodic Data Transfer FU*
- *Unsolicited Spontaneous Data Transfer FU*
- *Supervisory Control Data Transfer FU*

5.5.2.3 Invocation

5.5.2.3.1 Conditions préalables

Aucune invocation FU précédente ou concourante n'est nécessaire.

5.5.2.3.2 Restrictions

Dans un système INITIATOR donné, la *Permanent Association FU* ne doit pas être invoquée pour un système RESPONDER donné si:

- a) Une invocation de *Permanent Association FU* fonctionne déjà avec le même système RESPONDER, en utilisant des valeurs de paramètres identiques de la primitive *TASEBindReq*

ou:

- b) Une invocation de *Group Management FU* ou de *Group Definition FU* est en passe de fonctionner pour le même RESPONDER User Element¹⁸⁾.

5.5.2.3.3 Invoking events

Il est admis d'invoquer la partie INITIATOR de la *Permanent Association FU* uniquement par: – Une requête locale par le biais de la Coordinating Function, au démarrage du système.

De telles requêtes d'invocations locales doivent toujours être légales, c'est-à-dire que si la *Permanent Association FU* est présente, l'invocation de la *Permanent Association FU* considérée doit entrer dans sa boucle de rétablissement d'association, si l'association ne peut pas être établie immédiatement.

L'invocation de la partie RESPONDER de la *Permanent Association FU* est déclenchée par la réception d'une primitive *TASEBindInd* valide.

¹⁸⁾ L'objectif de cette règle est d'assurer l'intégrité du paramètre d'intégrité de configuration pour le système INITIATOR, dans le système RESPONDER. Ce paramètre est mis à jour à la fois par la *Group Management FU* et la *Group Definition FU* et est utilisé par la *Permanent Association FU*. Voir 5.4 .

5.5.2.2.4 Disrupted FUs

The *Permanent Association FU* may disrupt the following FUs, as a result of (temporary) association breakage:

- *Test Association FU*
- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*
- *Group Readout FU*
- *Requested Data Transfer FU*
- *Periodically Requested Data Transfer FU*
- *Unsolicited Data Transfer FU*
- *Unsolicited Periodic Data Transfer FU*
- *Unsolicited Spontaneous Data Transfer FU*
- *Supervisory Control Data Transfer FU*.

5.5.2.3 Invocation

5.5.2.3.1 Prerequisites

No previous or concurrent FU invocations are required.

5.5.2.3.2 Restrictions

In a given INITIATOR system, the *Permanent Association FU* shall not be invoked for a given RESPONDER system if:

- a) *Permanent Association FU* invocation is running with the same RESPONDER system, with identical values of the parameters in the *TASEBindReq* primitive
- or:
- b) *Group Management FU* invocation or a *Group Definition FU* invocation is currently being running for the same RESPONDER User Element¹⁸⁾.

5.5.2.3.3 Invoking events

The INITIATOR part of the *Permanent Association FU* may only be invoked by local request via the Coordinating Function, at system startup time.

Such local invocation requests shall always be legal, in the sense that if the *Permanent Association FU* is present, the *Permanent Association FU* invocation in question shall enter its association re-establishment loop if the association cannot be established immediately.

Invocation of the RESPONDER part of the *Permanent Association FU* is triggered by reception of a valid *TASEBindInd* primitive.

¹⁸⁾ The purpose of this rule is to ensure the integrity of the Configuration integrity parameter for the INITIATOR system in the RESPONDER system. This parameter is updated by both the *Group Management FU* and the *Group Definition FU* and utilized by the *Permanent Association FU*; see 5.4.

A noter que la partie RESPONDER de la *Permanent Association FU* doit être identique à la partie RESPONDER de la *Dynamic Association FU*¹⁹⁾, car aucune information de permanence explicite n'est véhiculée par la primitive *TASEBindInd*. En conséquence, il sera fait référence à la partie RESPONDER de la *Permanent Association FU* uniquement en tant que partie RESPONDER de l'*Association FU* tout au long de la partie restante de la description de la *Permanent Association FU*.

5.5.2.4 Fin

5.5.2.4.1 Orderly termination

La fin ordonnée de la *Permanent Association FU* n'est pas définie.

5.5.2.4.2 Disruption

Il est permis d'interrompre une invocation de *Permanent Association FU* au cours de la phase d'établissement, en recevant une primitive *TASEBindConf* dont le paramètre *Result* = *Rejected Permanent*²⁰⁾.

5.5.3 Procédures

5.5.3.1 Primitives de service TASE.1

Les services TASE.1 élémentaires suivants sont utilisés par la *Permanent Association FU*:

- *TASEBind*
- *TASEAbort*

5.5.3.1.1 Sequence

La séquence normale de primitives se divise en deux phases:

Phase 1: établissement de l'association

INITIATOR UE	TASE.1	RESPONDER UE
<i>TASEBindReq</i>	<-----	<i>TASEBindConf</i>
<i>TASEBindInd</i>	----->	<i>TASEBindResp</i>

Phase 2: fin brutale et rétablissement de l'association

INITIATOR UE	TASE.1	RESPONDER UE
<i>TASEAbortInd</i>	<-- -->	<i>TASEAbortInd</i>
<i>TASEBindReq</i>	<-----	<i>TASEBindConf</i>
<i>TASEBindInd</i>	----->	<i>TASEBindResp</i>

¹⁹⁾ En conséquence, étant donné que ceci est requis par la partie INITIATOR de la *Dynamic Association FU*, la partie RESPONDER de l'*Association FU* doit contenir un mécanisme de fin ordonnée, déclenché à partir de la partie INITIATOR. Cependant, la fin ordonnée ne sera jamais déclenchée par la partie INITIATOR d'une invocation *Permanent Association FU*, car aucun mécanisme de ce type n'est défini pour cette *FU*.

²⁰⁾ Toute autre tentative de rupture de la *Association FU* de la part du RESPONDER en terminant brutalement l'association aura finalement pour conséquence un rétablissement de l'association par la partie INITIATOR.

Note that the RESPONDER part of the *Permanent Association FU* necessarily must be identical to the RESPONDER part of the *Dynamic Association FU*¹⁹⁾, because no explicit information as to permanency is conveyed by the *TASEBindInd* primitive. Therefore, the RESPONDER part of the *Permanent Association FU* will be referred to as simply the RESPONDER part of the *Association FU* throughout the remaining part of this description of the *Permanent Association FU*.

5.5.2.4 Termination

5.5.2.4.1 Orderly termination

Orderly termination of the *Permanent Association FU* is not defined.

5.5.2.4.2 Disruption

A *Permanent Association FU* invocation may be disrupted in the establishment phase by receiving a *TASEBindConf* primitive with Result = *Rejected Permanent*²⁰⁾.

5.5.3 Procedures

5.5.3.1 TASE.1 service primitives

The following elementary TASE.1 services are used by the *Permanent Association FU*:

- *TASEBind*
- *TASEAbort*

5.5.3.1.1 Sequence

The normal sequence of primitives is partitioned into two phases:

Phase 1: establishing the association

INITIATOR UE	TASE.1	RESPONDER UE
<i>TASEBindReq</i>	<-----	<i>TASEBindConf</i>
<i>TASEBindInd</i>	----->	<i>TASEBindResp</i>

Phase 2: abruptly terminating and re-establishing the association

INITIATOR UE	TASE.1	RESPONDER UE
<i>TASEAbortInd</i>	<-- -->	<i>TASEAbortInd</i>
<i>TASEBindReq</i>	<-----	<i>TASEBindConf</i>
<i>TASEBindInd</i>	----->	<i>TASEBindResp</i>

¹⁹⁾ Consequently, since required by the INITIATOR part of the *Dynamic Association FU*, the RESPONDER part of the *Association FU* must contain a mechanism for Orderly Termination, triggered from the INITIATOR part. However, Orderly Termination will never be triggered by the INITIATOR part of a *Permanent Association FU* invocation, since no such mechanism is defined for this FU.

²⁰⁾ Any other attempt to disrupt the *Association FU* on the part of the RESPONDER by abruptly terminating the association will result in the INITIATOR part eventually re-establishing the association.

Règles:

- a) Une fois la phase 1 terminée, la phase 2 est susceptible d'intervenir à n'importe quel moment et un nombre illimité de fois (ou jamais)²¹⁾.
- b) Chaque fois que la phase 2 intervient, la primitive *TASEAbortInd* doit déclencher les actions suivantes:
 - Sur la partie INITIATOR:
 - Rupture locale de toutes les autres FU en cours de fonctionnement dans l'association.
 - Sur la partie RESPONDER:
 - Rupture locale de toutes les autres FU en cours de fonctionnement dans l'association.
 - Lors de l'établissement suivant de toute association ayant les caractéristiques nécessaires au transfert de données non sollicitées (à utiliser par la *Unsolicited Periodic Data Transfer FU* ou la *Unsolicited Spontaneous Data Transfer FU*) entre les membres de la même paire INITIATOR et RESPONDER UE, la partie RESPONDER de l'Association FU doit envoyer un code de redémarrage *Restart, spontaneous management lost*.
- c) Le délai entre la réception de la primitive *TASEAbortInd* et l'émission de la primitive *TASEBindReq* sur l'INITIATOR UE est défini au niveau local.
- d) Chaque fois que l'INITIATOR UE reçoit le code de redémarrage *Restart, spontaneous management lost* avec la primitive *TASEBindConf* en phase 1, la *Permanent Association FU* doit essayer d'invoquer la *Restart Reactivate FU* pour le RESPONDER considéré.
- e) Si une erreur de zone de contrôle de Configuration (voir 5.4) est détectée par l'INITIATOR UE avec la primitive *TASEBindConf*, en phase 1, la *Permanent Association FU* doit essayer d'invoquer la *Restart Reconfigure FU* pour le RESPONDER considéré.
- f) Les règles établies dans la procédure de vérification du Configuration Set (voir 5.4), sont également applicables.

5.5.3.1.2 Parameter values

TASEBind:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Initiator	Tel que spécifié dans la description de l'appel de <i>TASEBindReq</i> (voir annexe A de la CEI 60870-6-501). La «partie adresse» doit être égale à l'adresse du noeud de l'INITIATOR, telle qu'elle est spécifiée en 5.2. La «partie A-suffix» doit être comme spécifié en 5.2.1.	Copie de la valeur à partir de <i>Ind</i>
Acceptor	Tel que spécifié dans la description de l'appel de <i>TASEBindReq</i> (voir annexe A de la CEI 60870-6-501). La «partie adresse» doit être égale à l'adresse du noeud du RESPONDER, telle qu'elle est spécifiée en 5.2. La partie «A-suffix» doit être comme spécifié en 5.2.1.	Copie de la valeur à partir de <i>Ind</i>

²¹⁾ Contrairement à la *Dynamic Association FU*, toute occurrence de la primitive *TASEAbortInd* est traitée dans le cadre de l'invocation FU et ne constitue donc pas une erreur ni un événement de rupture de la *Permanent Association FU* ni une erreur.

Rules:

- a) Phase 2 may occur at any time after phase 1 has been terminated, and may occur any number of times, including none at all²¹⁾.
- b) Whenever phase 2 occurs, the *TASEAbortInd* primitive shall trigger the following actions:
 - In the INITIATOR part:
 - Local disruption of all other FUs that are running on the association.
 - In the RESPONDER part:
 - Local disruption of all other FUs that are running on the association.
 - The next time any association with the characteristics for Unsolicited Data Transfer (for use by the *Unsolicited Periodic Data Transfer FU* or *Unsolicited Spontaneous Data Transfer FU*) is established between the same INITIATOR and RESPONDER UE pair, the RESPONDER part of the *Association FU* shall signal restart code *Restart, spontaneous management lost*.
- c) The time delay between reception of the *TASEAbortInd* primitive and the issuing of the *TASEBindReq* primitive in the INITIATOR UE is a local matter.
- d) Whenever the restart code *Restart, spontaneous management lost* is received by the INITIATOR UE with the *TASEBindConf* primitive in phase 1, the *Permanent Association FU* shall try to invoke the *Restart Reactivate FU* for the RESPONDER in question.
- e) If Configuration control field error (see 5.4) is detected by the INITIATOR UE with the *TASEBindConf* primitive in phase 1, the *Permanent Association FU* shall try to invoke the *Restart Reconfigure FU* for the RESPONDER in question.
- f) The rules stated in the procedure for Configuration Set check (see 5.4) also apply.

5.5.3.1.2 Parameter values

TASEBind:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Initiator	As specified in description of <i>TASEBindReq</i> call in annex A of IEC 60870-6-501. The "address part" shall be equal to the address of the INITIATOR's own node as specified in 5.2. The "A-suffix" part shall be as specified in 5.2.1.	Copy of value from <i>Ind</i>
Acceptor	As specified in description of <i>TASEBindReq</i> call in annex A of IEC 60870-6-501. The "address part" shall be equal to the address of the RESPONDER's own node as specified in 5.2. The "A-suffix" part shall be as specified in 5.2.1.	Copy of value from <i>Ind</i>

²¹⁾ In contrast to the *Dynamic Association FU*, any occurrence of the *TASEAbortInd* primitive is handled within the FU invocation, and thus does not constitute a *Permanent Association FU* disruption event or error.

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
User-data	Octets, défini par l'utilisateur, facultatif.	Premier octet (octet n° 0): Facultatif Octets n° 1 à 16: Valeur Control Field. Pour obtenir des informations sur les règles de codage, voir 5.4.
	Le «security information field» commence à l'octet 2, s'il existe. La zone est codée comme décrit en 9.6.	Le «security information field» commence à l'octet 17 s'il existe. La zone est codée comme décrit en 9.6. Le «security information field» doit être présent si les classes de sécurité 2 et 3 sont utilisées. Il doit également être présent si la classe de sécurité 1 avec authentification sur deux voies est sélectionnée.
Result	(Non applicable)	Association établie: = <i>result-ok</i> Association non établie en raison d'une condition d'erreur: Autre valeur. Voir 5.5.3.2.

5.5.3.2 Traitement des erreurs

5.5.3.2.1 FU disruption

Il est permis d'interrompre la *Permanent Association FU* en lui envoyant une primitive *TASEBindConf* dont le paramètre Result = *Rejected Permanent* (voir ci-dessus).

5.5.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Permanent Association FU* n'est pas présente dans un INITIATOR UE:

Les requêtes d'invocation sont toujours générées localement au démarrage du système (voir section "Invocation" ci-dessus). Le traitement de ce type d'erreur se fait donc au niveau local et ne fait pas partie du domaine d'application de ce rapport.

L'*Association FU* doit être présente dans un RESPONDER UE pour que le système de communication soit en mesure de fonctionner.

FU présente, mais tentative illégale:

Les tentatives d'invocations multiples simultanées de la *Permanent Association FU* pour des jeux de paramètres identiques ne font pas partie du domaine d'application de ce rapport.

Les tentatives d'invocation de la *Permanent Association FU* en violant la règle 2 du 5.5.2.3.2 (*Group Management FU* ou *Group Definition FU* en cours de fonctionnement): Démarrer l'horloge de relance locale, et entrer dans la boucle de relance normale après expiration.

Parameter	Req (INITIATOR)	Resp (RESPONDER)
User-data	Octets, user defined, optional.	First octet (octet No. 0): Optional Octets No. 1 through 16: Control Field value. See the procedure for Configuration Set check, in 5.4, for coding rules.
	The security information field starts from octet 2 if present. The field is encoded as described in the 9.6.	The security information field starts from octet 17 if present. The field is encoded as described in 9.6. The security information field must be present if security classes 2 or 3 are used. It must be present if security class 1 with two-way authentication is selected, as well.
Result	(Not applicable)	Association established: = <i>result-ok</i> Association not established, due to error condition: Other value, see 5.5.3.2.

5.5.3.2 Error handling

5.5.3.2.1 FU disruption

The *Permanent Association FU* may be disrupted by receiving a *TASEBindConf primitive* with Result = *Rejected Permanent* (see above).

5.5.3.2.2 Illegal invocation attempt

FU not present:

If the *Permanent Association FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally at system startup time; see section "Invocation", above. Consequently, the handling of this type of error is a local issue, outside the scope of this report.

In a RESPONDER UE, the *Association FU* must be present, to make the communication system able to function at all.

FU present, but attempt illegal:

Attempts of multiple simultaneous invocations of the *Permanent Association FU* for identical parameter sets is a local issue, outside the scope of this report.

Attempted *Permanent Association FU* invocation violating rule 6 in 5.5.2.3.2 (*Group Management FU* or *Group Definition FU* running): Start local retry timer, and enter normal retry loop upon expiry.

5.5.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>TASEBindConf</i>
FU pas encore en fonctionnement	Ignorer ou indication d'erreur locale
FU en fonctionnement, en attente de <i>TASEBindConf</i>	(Normal)
En fonctionnement	Ignorer ou indication d'erreur locale/connexion
En attente d'expiration de l'horloge de relance	Ignorer ou indication d'erreur locale/connexion
NOTE – La primitive <i>TASEAbortInd</i> n'est jamais considérée comme hors contexte.	

Partie RESPONDER:

Etat	<i>TASEBindInd</i>
Inactif	(Normal)
En fonctionnement	Ignorer ou indication d'erreur locale/connexion
NOTE – La primitive <i>TASEAbortInd</i> n'est jamais considérée comme hors contexte.	

5.5.3.2.4 Timing errors

Aucune erreur d'horloge ne peut apparaître sur la partie INITIATOR.

Erreurs dans la partie RESPONDER:

Erreur	Réaction de la part de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à <i>TASEBindInd</i>	Sur la RESPONDER: <i>TASEAbortInd</i>, avec Reason = <i>misbehaviour-of-local-service-user</i>. Erreur locale lors d'une éventuelle tentative d'émission de <i>TASEBindResp</i> Sur l'INITIATOR: <i>TASEBindConf</i>, avec Result = <i>no-answer-from-remote-system</i>	Partie RESPONDER: Passer à l'état inactif. Partie INITIATOR: Voir 5.5.3.2.6.

5.5.3.2.5 Congestion error

Aucune erreur d'encombrement ne se produit dans la *Permanent Association FU*. Voir 4.2.1.

5.5.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>TASEBindConf</i>
FU not yet running	Ignore, or local error indication
FU running, waiting for <i>TASEBindConf</i>	(Normal)
Running	Ignore, or local error indication/logging
Waiting for retry timer expiry	Ignore, or local error indication/logging
NOTE – The <i>TASEAbortInd</i> primitive is never considered out of context.	

RESPONDER part:

State	<i>TASEBindInd</i>
Idle	(Normal)
Running	Ignore, or local error indication/logging
NOTE – The <i>TASEAbortInd</i> primitive is never considered out of context	

5.5.3.2.4 Timing errors

There can occur no timing errors in the INITIATOR part.

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>TASEBindInd</i>	In RESPONDER: <i>TASEAbortInd</i>, with Reason = <i>misbehaviour-of-local-service-user</i>. Local error from eventual attempt to issue <i>TASEBindResp</i> In INITIATOR: <i>TASEBindConf</i>, with Result = <i>no- answer-from-remote-system</i>.	RESPONDER part: Go to idle state. INITIATOR part: See 5.5.3.2.6.

5.5.3.2.5 Congestion error

Congestion errors do not occur in the *Permanent Association FU*. See 4.2.1.

5.5.3.2.6 TASE.1 service primitive parameter errors

Erreurs dans la primitive *TASEBindInd* (détectées par la partie RESPONDER):

Erreur	Action à effectuer dans la partie RESPONDER de la FU
Partie adresse du Initiator (identifiant le noeud INITIATOR) non valide, selon les tables d'accès locales	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>authentication-failure</i> . Passer à l'état inactif si l'état est différent.
Classe de sécurité requise par l'INITIATOR > 0, mais le RESPONDER ne supporte aucun mécanisme de sécurité	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>security-is-not-supported-by-service-user</i> . Passer à l'état inactif si l'état est différent.
Classe de sécurité requise par l'INITIATOR > 0, mais la longueur de la zone d'information d'authentification est zéro	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>Authentication-failure</i> . Passer à l'état inactif si l'état est différent.
L'INITIATOR a demandé une classe de sécurité ou des options de sécurité non supportées par le RESPONDER	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>incompatible-security-options-requested</i> . Passer à l'état inactif si l'état est différent.
L'INITIATOR a demandé une authentification de zone ou un cryptage pour un paramètre non présent dans la version TASE.1 sélectionné	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>incompatible-security-options-requested</i> . Passer à l'état inactif si l'état est différent.
L'INITIATOR a demandé une option de sécurité (authentification de zone ou cryptage) non supportée par le RESPONDER	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>incompatible-security-requested</i> . Passer à l'état inactif si l'état est différent.
Le RESPONDER n'a pas pu authentifier l'INITIATOR à partir de la zone d'information d'authentification reçue	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>authentication-failure</i> . Passer à l'état inactif si l'état est différent.

Erreurs dans la primitive *TASEBindConf* (détectées par la partie INITIATOR):

Erreur	Action à effectuer dans la partie INITIATOR de la FU
Zéro octet dans le User-data, ou marquage de redémarrage dans User-data non défini en 5.3 et Result = <i>result-ok</i> .	Traiter comme User-data. normal PAS une erreur; passer à l'état de fonctionnement comme normal.
Result >< <i>result-ok</i>	Démarrer l'horloge de relance locale et entrer dans la boucle de relance normale après expiration ²²⁾

5.6 Dynamic Association FU

Type: Primary.

5.6.1 Fonction

Une invocation de la *Dynamic Association FU* fournit les séquences fonctionnelles suivantes:

- L'INITIATOR UE établit une association avec un RESPONDER UE.
- Si, au cours de l'établissement de l'association, le RESPONDER UE a demandé à l'INITIATOR UE de le réinitialiser, l'INITIATOR UE invoque la FU correspondante.

²²⁾ Si les classes de sécurité sélectionnées sont 1-3, il convient de considérer attentivement la stratégie de relance. Il n'est pas conseillé d'entrer une relance normale. Ceci donne éventuellement au récepteur de nouvelles occasions de recevoir des zones d'authentification utilisables pour un redémarrage/masquage ultérieur ou pour des tentatives de rupture de clés de cryptage.

5.5.3.2.6 TASE.1 service primitive parameter errors

Errors in the *TASEBindInd* primitive (detected by the RESPONDER part):

Error	Action in RESPONDER part of FU
Address part of the Initiator (identifying the INITIATOR node) not valid, according to local access tables	Issue <i>TASEBindResp</i> primitive with Result = <i>authentication-failure</i> . Enter idle state, if in other state.
INITIATOR requested security class > 0, but the RESPONDER does not support any security mechanism	Issue <i>TASEBindResp</i> primitive with Result = <i>security-is-not-supported-by-service-user</i> . Enter idle state, if in other state.
INITIATOR requested security class > 0, but the length of the authentication information field is zero	Issue <i>TASEBindResp</i> primitive with Result = <i>authentication-failure</i> . Enter idle state, if in other state.
INITIATOR requested a security class or security options not supported by the RESPONDER	Issue <i>TASEBindResp</i> primitive with Result = <i>incompatible-security-options-requested</i> . Enter idle state, if in other state.
INITIATOR requested field authentication or encipherment for a parameter not present in the selected TASE.1 version	Issue <i>TASEBindResp</i> primitive with Result = <i>incompatible-security-options-requested</i> . Enter idle state, if in other state.
INITIATOR requested a security option (field authentication or encipherment) not supported by the RESPONDER	Issue <i>TASEBindResp</i> primitive with Result = <i>incompatible-security-options-requested</i> . Enter idle state, if in other state.
RESPONDER could not authenticate the INITIATOR on the basis of the received authentication information field	Issue <i>TASEBindResp</i> primitive with Result = <i>authentication-failure</i> . Enter idle state, if in other state.

Errors in the *TASEBindConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Zero octets in User-data, or restart marking in User-data not defined in table in 5.3, and Result = <i>result-ok</i> .	Treat as normal User-data. NOT an error; enter running state as normal.
Result >< <i>result-ok</i>	Start local retry timer, and enter normal retry loop upon expiry ²²⁾ .

5.6 Dynamic Association FU

Type: Primary.

5.6.1 Function

A *Dynamic Association FU* invocation provides the following functional sequence:

- The INITIATOR UE establishes an association with a RESPONDER UE.
- If the RESPONDER UE has requested a re-initialization function from the INITIATOR UE during association establishment, the corresponding FU is invoked in the INITIATOR UE.

²²⁾ If security classes 1-3 are selected, the retry strategy should be carefully considered. It may not be advisable to enter normal retry. This may give the receiver new opportunities to receive authentication fields which may be used for subsequent replay/masquerade or for attempts to break the encipherment keys.

- c) Chaque fois que la *Dynamic Association FU* enregistre ensuite l'association comme brisée (interrompue brutalement), l'invocation de la *Dynamic Association FU* est interrompue.
- d) Au cours de la fin ordonnée de la *Dynamic Association FU*, l'INITIATOR UE libère l'association.

5.6.2 Règles de coordination

5.6.2.1 Utilisation de l'association

L'invocation de la FU crée et libère elle-même une association. Aucune autre association n'est utilisée par cette invocation de FU.

5.6.2.2 Relation avec les autres FU

5.6.2.2.1 Invoking FUs

Il est admis que la FU suivante invoque la *Dynamic Association FU*:

- *Restart Reconfigure FU*

5.6.2.2.2 Invoked FUs

Il est permis à la *Dynamic Association FU* d'invoquer les FU suivantes:

- *Restart Reconfigure FU*

5.6.2.2.3 Disrupting FUs

Il n'est permis à aucune autre FU d'interrompre directement la *Dynamic Association FU*.

Il est permis aux FU suivantes d'interrompre indirectement la *Dynamic Association FU* en déclenchant une fin d'association brutale:

- *Test Association FU*
- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*
- *Group Readout FU*
- *Requested Data Transfer FU*
- *Periodically Requested Data Transfer FU*
- *Supervisory Control Data Transfer FU*

5.6.2.2.4 Disrupted FUs

Il est permis à la *Dynamic Association FU* d'interrompre les FU suivantes, à la suite d'une rupture de la *Dynamic Association FU* elle-même:

- *Test Association FU*
- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*
- *Group Readout FU*
- *Requested Data Transfer FU*
- *Periodically Requested Data Transfer FU*
- *Supervisory Control Data Transfer FU*

- c) At any time later, whenever the *Dynamic Association FU* registers the association as being broken (abruptly terminated), the *Dynamic Association FU* invocation will be disrupted.
- d) During orderly termination of the *Dynamic Association FU* invocation, the INITIATOR UE releases the association.

5.6.2 Coordination rules

5.6.2.1 Association usage

The FU invocation itself creates and releases an association. No other association is used by that FU invocation.

5.6.2.2 Relation to other FUs

5.6.2.2.1 Invoking FUs

The *Dynamic Association FU* may be invoked by the following FU:

- *Restart Reconfigure FU*.

5.6.2.2.2 Invoked FUs

The *Dynamic Association FU* may invoke the following FUs:

- *Restart Reconfigure FU*.

5.6.2.2.3 Disrupting FUs

The *Dynamic Association FU* may not be disrupted directly by any other FU.

Indirectly the *Dynamic Association FU* may be disrupted by any of the following FUs triggering abrupt association termination:

- *Test Association FU*
- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*
- *Group Readout FU*
- *Requested Data Transfer FU*
- *Periodically Requested Data Transfer FU*
- *Supervisory Control Data Transfer FU*

5.6.2.2.4 Disrupted FUs

The *Dynamic Association FU* may disrupt the following FUs, as a result of disruption of the *Dynamic Association FU* itself:

- *Test Association FU*
- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*
- *Group Readout FU*
- *Requested Data Transfer FU*
- *Periodically Requested Data Transfer FU*
- *Supervisory Control Data Transfer FU*

5.6.2.3 Invocation

5.6.2.3.1 Conditions préalables

Aucune invocation de FU précédente ou concourante n'est requise.

5.6.2.3.2 Restrictions

Dans un système INITIATOR donné, la *Dynamic Association FU* ne doit pas être invoquée pour un système RESPONDER donné si:

a) Une invocation de la *Dynamic Association FU* fonctionne déjà avec le même système RESPONDER, en utilisant des valeurs de paramètres identiques de la primitive *TASEBindReq*;

ou:

b) Une invocation de la *Group Management FU* ou de la *Group Definition FU* est en cours de fonctionnement pour le même RESPONDER²³⁾.

5.6.2.3.3 Invoking events

Il est admis que l'invocation de la partie INITIATOR de la *Dynamic Association FU* se fasse par:

- une requête locale par le biais de la Coordinating Function, dont l'origine est en dehors du domaine d'application de ce rapport;
- la *Restart Reconfigure FU* (qui, à son tour, a été invoquée par une autre invocation de la *Dynamic Association FU* ou de la *Permanent Association FU* contrôlant une association différente).

L'invocation de la partie RESPONDER de la *Dynamic Association FU* est déclenchée par la réception d'une primitive *TASEBindInd* valide.

Ainsi que l'explique la section correspondante de la description de la *Permanent Association FU*, la partie RESPONDER de la *Dynamic Association FU* est identique à la partie RESPONDER de la *Permanent Association FU*. Il sera donc fait référence à la partie RESPONDER de la *Dynamic Association FU* uniquement en tant que partie RESPONDER de l'*Association FU* tout au long de la partie restante de la description de la *Dynamic Association FU*.

5.6.2.4 Fin

5.6.2.4.1 Orderly termination

La fin ordonnée de la partie INITIATOR d'une invocation de la *Dynamic Association FU* est déclenchée par une requête locale par le biais de la Coordinating Function, dont l'origine est:

- une fonction de niveau supérieur, qui ne fait pas partie du domaine d'application du présent rapport technique²⁴⁾;
- l'invocation de la *Restart Reconfigure FU* qui a invoqué la *Dynamic Association FU*.

La fin ordonnée de la partie RESPONDER d'une invocation de l'*Association FU* est déclenchée par la réception d'une primitive *A-ReleaseInd*.

²³⁾ L'objectif de cette règle est d'assurer l'intégrité du paramètre d'intégrité de la configuration, qui est mis à jour à la fois par la *Group Management FU* et la *Group Definition FU* et utilisé par la *Dynamic Association FU*.

²⁴⁾ Le présent rapport technique n'impose donc aucune limite quant à la stratégie adoptée par l'INITIATOR UE pour déterminer le moment auquel une association dynamique sera libérée.

5.6.2.3 Invocation

5.6.2.3.1 Prerequisites

No previous or concurrent FU invocations are required.

5.6.2.3.2 Restrictions

In a given INITIATOR system, the *Dynamic Association FU* shall not be invoked for a given RESPONDER system if:

a) a *Dynamic Association FU invocation* is already running with the same RESPONDER system, with identical values of the parameters in the *TASEBindReq* primitive;

or:

b) a *Group Management FU* invocation or a *Group Definition FU* invocation is running for the same RESPONDER²³⁾.

5.6.2.3.3 Invoking events

The INITIATOR part of the *Dynamic Association FU* may be invoked by:

- local request via the Coordinating Function, the original source of which is outside the scope of this report;
- The *Restart Reconfigure FU* (which in turn has been invoked by another *Dynamic Association FU* or *Permanent Association FU* invocation controlling a different association).

Invocation of the RESPONDER part of the *Dynamic Association FU* is triggered by reception of a valid *TASEBindInd* primitive.

As explained in the corresponding section of the *Permanent Association FU* description, the RESPONDER part of the *Dynamic Association FU* is identical to the RESPONDER part of the *Permanent Association FU*. Therefore, the RESPONDER part of the *Dynamic Association FU* will be referred to as simply the RESPONDER part of the *Association FU* throughout the remaining part of this description of the *Dynamic Association FU*.

5.6.2.4 Termination

5.6.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of a *Dynamic Association FU* invocation may be triggered by a local request via the Coordinating Function, originating from:

- a higher-level function, which is outside the scope of this technical report²⁴⁾
- the *Restart Reconfigure FU* invocation that invoked the *Dynamic Association FU*.

Orderly termination of the RESPONDER part of an *Association FU* invocation is triggered by reception of an *A-ReleaseInd* primitive.

²³⁾ The purpose of this rule is to ensure the integrity of the Configuration integrity parameter, which is updated by both the *Group Management FU* and the *Group Definition FU* and utilized by the *Dynamic Association FU*.

²⁴⁾ Consequently, this technical report places no restriction on the strategy adopted by the INITIATOR UE for determining the point in time at which a dynamic association will be released.

5.6.2.4.2 Disruption

Il est permis d'interrompre l'invocation de la *Dynamic Association FU* par:

- une erreur fatale dans l'invocation de toute autre FU actuellement supportée par l'invocation de la *Dynamic Association FU*. Voir 5.6.2.2.3;
- l'invocation de la *Dynamic Association FU* elle-même, après une erreur fatale. Voir 5.6.3.2.

5.6.3 Procédures

5.6.3.1 Primitives de service TASE.1

Les services TASE.1 élémentaires suivants sont utilisés par la *Dynamic Association FU*:

- *TASEBind*
- *TASEAbort*
- *TASEUnbind*

5.6.3.1.1 Sequence

La séquence normale de primitives est divisée en deux phases:

Phase 1: établissement de l'association

INITIATOR UE	TASE	RESPONDER UE
<i>TASEBindReq</i>	----->	<i>TASEBindInd</i>
<i>TASEBindConf</i>	<-----	<i>TASEBindResp</i>

Phase 2: fin ordonnée: libération de l'association

INITIATOR UE	TASE	RESPONDER UE
<i>TASEUnbindReq</i>	----->	<i>TASEUnbindInd</i>
<i>TASEUnbindConf</i>	<-----	<i>TASEUnbindResp</i>

Règles:

- a) Toute primitive *TASEAbortInd* reçue par l'INITIATOR UE dans l'association, en conséquence de la séquence de phase 1 (voir ci-dessus), doit déclencher localement:
 - La rupture de l'invocation de la *Dynamic Association FU*²⁵⁾.
 - La rupture de toutes les autres FU en cours de fonctionnement dans l'association.
- b) Toute primitive *TASEAbortInd* reçue par le RESPONDER UE dans l'association, à la suite de la séquence de phase 1 (voir ci-dessus), doit déclencher localement:
 - La rupture de l'invocation de l'Association FU²⁶⁾.
 - La rupture de toutes les autres FU en cours de fonctionnement dans l'association.

²⁵⁾ Contrairement à la *Permanent Association FU*, la *Dynamic Association FU* considère la primitive *TASEAbortInd* comme un événement d'erreur.

²⁶⁾ Cette rupture ne se distingue pas de l'action normale effectuée dans l'Association FU du RESPONDER lorsque l'association est contrôlée par une invocation de la *Permanent Association FU* sur l'INITIATOR.

5.6.2.4.2 Disruption

An invocation of the *Dynamic Association FU* may be disrupted by:

- fatal error in an invocation of any other FU currently being supported by the *Dynamic Association FU* invocation. See 5.6.2.2.3;
- the *Dynamic Association FU* invocation itself, upon fatal error. See 5.6.3.2.

5.6.3 Procedures

5.6.3.1 TASE.1 service primitives

The following elementary TASE.1 services are used by the *Dynamic Association FU*:

- *TASEBind*
- *TASEAbort*
- *TASEUnbind*

5.6.3.1.1 Sequence

The normal sequence of primitives is partitioned into two phases:

Phase 1: establishing the association

INITIATOR UE	TASE	RESPONDER UE
<i>TASEBindReq</i>	----->	<i>TASEBindInd</i>
<i>TASEBindConf</i>	<-----	<i>TASEBindResp</i>

Phase 2: orderly termination: releasing the association

INITIATOR UE	TASE	RESPONDER UE
<i>TASEUnbindReq</i>	----->	<i>TASEUnbindInd</i>
<i>TASEUnbindConf</i>	<-----	<i>TASEUnbindResp</i>

Rules:

- a) Any *TASEAbortInd* primitive received by the INITIATOR UE on the association, subsequent to the phase 1 sequence above, shall trigger, locally:
 - Disruption of the *Dynamic Association FU* invocation ²⁵⁾.
 - Disruption of all other FUs that are running on the association.
- b) Any *TASEAbortInd* primitive received by the RESPONDER UE on the association, subsequent to the phase 1 sequence above, shall trigger, locally:
 - Disruption of the *Association FU* invocation ²⁶⁾.
 - Disruption of all other FUs that are running on the association.

²⁵⁾ Unlike the *Permanent Association FU*, the *Dynamic Association FU* considers the *TASEAbortInd* primitive an error event.

²⁶⁾ This disruption is indistinguishable from the normal action performed in the *Association FU* in the RESPONDER when the association is controlled by a *Permanent Association FU* invocation in the INITIATOR.

- c) La durée écoulée entre la fin de la phase 1 et le début de la phase 2 est définie localement dans l'INITIATOR et ne fait pas partie du domaine d'application de ce rapport technique²⁷⁾.
- d) Chaque fois que l'INITIATOR UE reçoit un marquage de redémarrage *Restart, spontaneous management lost* avec la primitive *TASEBindConf*, en phase 1, la *Dynamic Association FU* doit essayer d'invoquer la *Restart Reactivate FU* pour le RESPONDER considéré.
- e) Si une erreur de zone de contrôle de Configuration²⁸⁾ est détectée par l'INITIATOR UE avec la primitive *TASEBindConf* en phase 1, la *Dynamic Association FU* doit essayer d'invoquer la *Restart Reconfigure FU* pour le RESPONDER considéré.
- f) Les règles établies dans les procédures de vérification du Configuration Set, en 5.4, s'appliquent également.

5.6.3.1.2 Parameter values

TASEBind:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Initiator	Tel que spécifié dans la description de l'appel de <i>TASEBindReq</i> (voir annexe A de la CEI 60870-6-501). La "partie adresse" doit être égale à l'adresse du noeud de l'INITIATOR, telle qu'elle est spécifiée en 5.2. La "partie A-suffix" doit être telle qu'elle est spécifiée en 5.2.1.	Copie de la valeur à partir de <i>Ind</i>
Acceptor	Tel que spécifié dans la description de l'appel de <i>TASEBindReq</i> (voir annexe A de la CEI 60870-6-501). La "partie adresse" doit être égale à l'adresse du noeud du RESPONDER, telle qu'elle est spécifiée en 5.2. La partie "A-suffix" doit être telle qu'elle est spécifiée en 5.2.1.	Copie de la valeur à partir de <i>Ind</i>
User-data	Toute valeur autorisée par la TAPI ²⁹⁾	Premier octet (octet n° 0): Facultatif (voir ci-dessus) Octets n° 1 à 16: Valeur Control Field. Pour obtenir des informations sur les règles de codage, voir 5.4.
	Le «security information field» commence à l'octet 2, s'il existe. La zone est codée comme décrit en 9.6.	Le «security information field» commence à l'octet 17 s'il existe. La longueur maximale est de 66 octets. La zone est codée comme décrit en 9.6. Le «security information field» doit être présent si les classe de sécurité 2 ou 3 sont utilisées. Il doit également être présent si la classe de sécurité 1 avec authentification sur deux voies est sélectionnée.

²⁷⁾ Trois stratégies sont possibles:

- a) Entrer en phase 2 immédiatement, s'il n'y a plus de trafic.
- b) Entrer en phase 2 après un délai fixe, lorsqu'il n'y a plus de trafic.
- c) Entrer en phase 2 uniquement après l'action de l'opérateur.

²⁸⁾ Voir 5.4.

²⁹⁾ Les transferts de données non sollicitées et périodiques n'interviennent jamais par le biais d'associations définies par la *Dynamic Association FU* mais uniquement par la *Permanent Association FU*.

- c) The time span between end of phase 1 and start of phase 2 is a local issue in the INITIATOR, outside the scope of this technical report²⁷⁾.
- d) Whenever the restart marking *Restart, spontaneous management lost* is received by the INITIATOR UE with the *TASEBindConf* primitive in phase 1, the *Dynamic Association FU* shall try to invoke the *Restart Reactivate FU* for the RESPONDER in question.
- e) If Configuration control field error²⁸⁾ is detected by the INITIATOR UE with the *TASEBindConf* primitive in phase 1, the *Dynamic Association FU* shall try to invoke the *Restart Reconfigure FU* for the RESPONDER in question.
- f) The rules stated in the procedure for Configuration Set check in 5.4 also apply.

5.6.3.1.2 Parameter values

TASEBind:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Initiator	As specified in description of <i>TASEBindReq</i> call in annex A of IEC 60870-6-501. The "Address part" shall be equal to the address of the INITIATOR's own node as specified in 5.2. The "A-suffix" part shall be as specified in 5.2.1.	Copy of value from <i>Ind</i>
Acceptor	As specified in description of <i>TASEBindReq</i> call in annex A of IEC 60870-6-501. The "Address part" shall be equal to the address of the RESPONDER's own node as specified in 5.2. The "A-suffix" part shall be as specified in 5.2.1.	Copy of value from <i>Ind</i>
User-data	Any value allowed by the TAPI ²⁹⁾ .	First octet (octet No. 0): Optional in the above. Octets No. 1 through 16: Control Field value. See the procedure for Configuration Set check, in 5.4 for coding rules.
	The security information field starts from octet 2 if present. The field is encoded as described in 9.6.	The security information field starts from octet 17 if present. The maximum length is 66 octets. The field is encoded as described in 9.6. The security information field must be present if security classes 2 or 3 are used. It must be present if security class 1 with two-way authentication is selected, as well.

²⁷⁾ Three possible strategies:

- a) Enter phase 2 immediately, when there is no more traffic.
- b) Enter phase 2 after a fixed delay time after the occurrence of no more traffic.
- c) Enter phase 2 only after operator action.

²⁸⁾ See 5.4.

²⁹⁾ Unsolicited and periodic data transfers never occur via associations set up by the *Dynamic Association FU*; only by the *Permanent Association FU*.

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Result	(Non applicable)	Association établie: = <i>result-ok</i> Association non établie en raison d'une condition d'erreur: Autre valeur. Voir 5.6.3.2. (Ceci signifie que la partie RESPONDER de la Association FU n'a pas été invoquée pour l'association.)

TASEUnbind:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
User-reason	Tel que spécifié dans la description de l'appel de <i>TASEUnbindReq</i> (voir annexe A de la CEI 60870-6-501); ce choix ne fait pas partie du domaine d'application de ce rapport.	(Non applicable)
Result	(Non applicable)	Toujours égal à <i>result-ok</i>

5.6.3.2 Traitement des erreurs

5.6.3.2.1 FU disruption

La rupture par une autre FU n'est effectuée qu'indirectement:

Chaque fois qu'une invocation de la *Dynamic Association FU* est interrompue, elle doit à son tour déclencher une rupture directe de toute invocation de FU qui utilise l'association contrôlée par la *Dynamic Association FU* interrompue.

Les procédures de nettoyage locales ne sont pas spécifiées par le présent rapport technique.

5.6.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Dynamic Association FU* n'est pas présente dans un INITIATOR UE:

Le traitement, par le biais de la Coordinating Function, des requêtes d'invocations locales ayant échoué, se fait localement, et ne fait pas partie du domaine d'application du présent rapport technique.

Echec des tentatives d'invocation par la *Restart Reconfigure FU*: Voir la description de la *Restart Reconfigure FU*.

Dans un RESPONDER UE, il faut que l'*Association FU* soit présente pour pouvoir fonctionner.

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Result	(Not applicable)	Association established: = <i>result-ok</i> Association not established, due to error condition: Other value. See 5.6.3.2. (This means that the RESPONDER part of the <i>Association FU</i> has not been invoked, for the association).

TASEUnbind:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
User-reason	As specified in description of <i>TASEUnbindReq</i> call in annex A of IEC 60870-6-501. Choice outside scope of this report.	(Not applicable)
Result	(Not applicable)	Always equal to <i>result-ok</i>

5.6.3.2 Error handling

5.6.3.2.1 FU disruption

Disruption by another FU is only done indirectly:

Whenever a *Dynamic Association FU* invocation is disrupted, it shall in turn trigger direct disruption of any FU invocations that use the association controlled by the disrupted *Dynamic Association FU*.

Local clean-up procedures are not specified by this technical report.

5.6.3.2.2 Illegal invocation attempt

FU not present:

If the *Dynamic Association FU* is not present in an INITIATOR UE:

The handling of failed local invocation requests via the Coordinating Function is a local issue, outside the scope of this technical report.

Failing invocation attempts by the *Restart Reconfigure FU*: see description of the *Restart Reconfigure FU*.

In a RESPONDER UE, the *Association FU* must be present to be able to function at all.

FU présente, mais tentative illégale:

Les tentatives d'invocations multiples simultanées de la *Dynamic Association FU* pour des jeux de paramètres identiques par le biais de la Coordinating Function sont traitées localement et ne font pas partie du domaine d'application du présent rapport technique.

S'il s'agit d'une tentative d'invocation de la *Dynamic Association FU* violant la règle b) du 5.6.2.3.2 (*Group Management FU* ou *Group Definition FU* en cours de fonctionnement): Renvoyer l'erreur à la Coordinating Function locale.

En cas d'échec de tentatives d'invocation par la *Restart Reconfigure FU*: Voir la description de la *Restart Reconfigure FU*.

5.6.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>TASEBindConf</i>	<i>TASEUnbindConf</i>
FU pas en fonctionnement	Ignorer, ou indication d'erreur locale	Ignorer, ou indication d'erreur locale
FU en fonctionnement, en attente d'une <i>TASEBindConf</i>	(Normal)	Passer à l'état inactif
En cours de fonctionnement	Ignorer, ou indication d'erreur locale/connexion	Passer à l'état inactif
En attente d'une <i>TASEUnbindConf</i>	Passer à l'état inactif	(Normal)
NOTE – La primitive <i>TASEAbortInd</i> n'est jamais considérée comme hors contexte.		

Partie RESPONDER:

Etat	<i>TASEBindInd</i>	<i>TASEUnbindInd</i>
Inactif	(Normal)	Emettre la <i>TASEUnbindResp</i> normale
En fonctionnement	Ignorer, ou indication d'erreur locale/connexion	(Normal)
NOTE – La primitive <i>TASEAbortInd</i> n'est jamais considérée comme hors contexte.		

5.6.3.2.4 Timing errors

Aucune erreur d'horloge ne peut se produire dans la partie INITIATOR.

Erreurs dans la partie RESPONDER:

Erreur	Réaction de la part de TASE.1	Action spécifiée dans FU
Retard de l'UE dans sa réponse à la <i>TASEBindInd</i>	Sur le RESPONDER: <i>TASEAbortInd</i>, avec Reason = <i>misbehaviour-of-local-service-user</i>. Erreur locale lors de la tentative éventuelle d'émission de <i>TASEBindResp</i> Sur l'INITIATOR: <i>TASEBindConf</i>, avec Result = <i>no-answer-from-remote-system</i>	Partie RESPONDER: Passer à l'état inactif. Partie INITIATOR: Voir 5.6.3.2.6.

FU present, but attempt illegal:

Attempts of multiple simultaneous invocations of the *Dynamic Association FU* for identical parameter sets via the Coordinating Function is a local issue, outside the scope of this technical report.

Attempted *Dynamic Association FU* invocation attempt violating rule b) in 5.6.2.3.2 (*Group Management FU* or *Group Definition FU* running): report the error back to the local Coordinating Function.

Failing invocation attempts by the *Restart Reconfigure FU*: see description of the *Restart Reconfigure FU*.

5.6.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>TASEBindConf</i>	<i>TASEUnbindConf</i>
FU not running	Ignore, or local error indication	Ignore, or local error indication
FU running, waiting for <i>TASEBindConf</i>	(Normal)	Go to idle state
Running	Ignore, or local error indication/logging	Go to idle state
Waiting for <i>TASEUnbindConf</i>	Go to idle state	(Normal)
NOTE – The <i>TASEAbortInd</i> primitive is never considered out of context.		

RESPONDER part:

State	<i>TASEBindInd</i>	<i>TASEUnbindInd</i>
Idle	(Normal)	Issue normal <i>TASEUnbindResp</i>
Running	Ignore, or local error indication/logging	(Normal)
NOTE – The <i>TASEAbortInd</i> primitive is never considered out of context.		

5.6.3.2.4 Timing errors

There can occur no timing errors in the INITIATOR part.

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>TASEBindInd</i>	In RESPONDER: <i>TASEAbortInd</i>, with Reason = <i>misbehaviour-of-local-service-user</i>. Local error from eventual attempt at issuing <i>TASEBindResp</i> In INITIATOR: <i>TASEBindConf</i>, with Result = <i>no-answer-from-remote-system</i>.	RESPONDER part: Go to idle state. INITIATOR part: See 5.6.3.2.6.

Erreur	Réaction de la part de TASE.1	Action spécifiée dans FU
Retard de l'UE dans sa réponse à la <i>TASEUnbindInd</i>	Sur le RESPONDER: <i>TASEAbortInd</i>, avec Reason = <i>misbehaviour-of-local-service-user</i>. Erreur locale lors de la tentative d'émission de la <i>TASEUnbindResp</i> Sur l'INITIATOR: <i>TASEAbortInd</i>, avec Reason = <i>no-answer-from-remote-system</i>	Partie RESPONDER: Traiter comme une fin brutale d'association. Partie INITIATOR: Traiter comme une fin brutale d'association.

5.6.3.2.5 Congestion error

Aucune erreur d'encombrement ne se produit dans la *Dynamic Association FU*. Voir 4.2.1.

5.6.3.2.6 TASE.1 service primitive parameter errors

Erreurs dans la primitive *TASEBindInd* (détectées par la partie RESPONDER):

Erreur	Action dans la partie RESPONDER de la FU
Partie de niveau bas de l'INITIATOR (identifiant le noeud INITIATOR) non valide, selon les tables d'accès locales	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>authentication-failure</i> . Passer à l'état inactif si l'état est différent.
Classe de sécurité requise par l'INITIATOR > 0, mais le RESPONDER ne supporte aucun mécanisme de sécurité	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>security-is-not-supported-by-service-user</i> Passer à l'état inactif si l'état est différent.
Classe de sécurité requise par l'INITIATOR > 0, mais la longueur de la zone d'information d'authentification est zéro	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>authentication-failure</i> . Passer à l'état inactif si l'état est différent.
L'INITIATOR a demandé une classe de sécurité ou des options de sécurité non supportées par le RESPONDER	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>incompatible-security-options-requested</i> . Passer à l'état inactif si l'état est différent.
L'INITIATOR a demandé une authentification de zone ou un cryptage pour un paramètre non présent dans la classe ou la version ELCOM sélectionnée	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>incompatible-security-options-requested</i> . Passer à l'état inactif si l'état est différent.
L'INITIATOR a demandé une option de sécurité (authentification de zone ou cryptage) non supportée par le RESPONDER	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>incompatible-security-options-requested</i> . Passer à l'état inactif si l'état est différent.
Le RESPONDER n'a pas pu authentifier l'INITIATOR à partir de la zone d'information d'authentification reçue	Emettre la primitive <i>TASEBindResp</i> avec Result = <i>authentication-failure</i> Passer à l'état inactif si l'état est différent.

A noter que User-data est toujours valide dans la *Dynamic Association FU*, à condition qu'il soit valide dans la TAPI.

Erreurs dans la primitive *TASEUnbindInd* (détectées par la partie RESPONDER):

La seule erreur possible est une valeur User-reason illégale, qui est considérée comme hors du domaine d'application de ce rapport.

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>TASEUnbindInd</i>	In RESPONDER: <i>TASEAbortInd</i> , with Reason = <i>misbehaviour-of-local-service-user</i> . Local error from any attempt at issuing <i>TASEUnbindResp</i> In INITIATOR: <i>TASEAbortInd</i> , with Reason = <i>no-answer-from-remote-system</i>	RESPONDER part: Handle as abrupt termination of association. INITIATOR part: Handle as abrupt termination of association.

5.6.3.2.5 Congestion error

Congestion errors do not occur in the *Dynamic Association FU*. See 4.2.1.

5.6.3.2.6 TASE.1 service primitive parameter errors

Errors in the *TASEBindInd* primitive (detected by the RESPONDER part):

Error	Action in RESPONDER part of FU
Lower level part of Initiator (identifying the INITIATOR node) not valid, according to local access tables	Issue <i>TASEBindResp</i> primitive with Result = <i>authentication-failure</i> . Enter idle state, if in other state.
INITIATOR requested security class > 0, but the RESPONDER does not support any security mechanism	Issue <i>TASEBindResp</i> primitive with Result = <i>security-is-not-supported-by-service-user</i> . Enter idle state, if in other state.
INITIATOR requested security class > 0, but the length of the authentication information field is zero.	Issue <i>TASEBindResp</i> primitive with Result = <i>authentication-failure</i> . Enter idle state, if in other state.
INITIATOR requested a security class or security options not supported by the RESPONDER	Issue <i>TASEBindResp</i> primitive with Result = <i>incompatible-security-options-requested</i> . Enter idle state, if in other state.
INITIATOR requested field authentication or encipherment for a parameter not present in the selected ELCOM class or version	Issue <i>TASEBindResp</i> primitive with Result = <i>incompatible-security-options-requested</i> . Enter idle state, if in other state.
INITIATOR requested a security option (field authentication or encipherment) not supported by the RESPONDER	Issue <i>TASEBindResp</i> primitive with Result = <i>incompatible-security-options-requested</i> . Enter idle state, if in other state.
RESPONDER could not authenticate the INITIATOR on the basis of the received authentication information field	Issue <i>TASEBindResp</i> primitive with Result = <i>authentication-failure</i> . Enter idle state, if in other state.

Note that User-data is always valid within the *Dynamic Association FU*, provided it is valid within the TAPI.

Errors in the *TASEUnbindInd* primitive (detected by the RESPONDER part):

The only error possible is illegal User-reason value, which is considered outside the scope of this report.

Erreurs dans la primitive *TASEBindConf* (détectées par la partie INITIATOR):

Erreur	Action à effectuer dans la partie INITIATOR de la FU
Initiator >< Initiator dans la primitive <i>TASEBindReq</i> correspondante, et Result = <i>result-ok</i> . ³⁰⁾ (mélange d'entités INITIATOR FU dans le RESPONDER.)	Renvoyer l'erreur à la Coordinating Function et déclencher une fin brutale locale d'association.
Zéro octet dans le User-data, et Result = <i>result-ok</i> .	Traiter comme User-data normal. PAS une erreur; passer à l'état de fonctionnement comme normal.
Result >< <i>result-ok</i>	Renvoyer l'erreur à la Coordinating Function et passer à l'état inactif.

Erreurs dans la primitive *TASEUnbindConf* (détectées par la partie INITIATOR):

Erreur	Action sur la partie INITIATOR de la FU
Result >< <i>result-ok</i>	Renvoyer l'erreur à la Coordinating Function et déclencher une fin brutale locale d'association.

5.7 Test Association FU

Type: Primary.

5.7.1 Fonction

Une invocation de la *Test Association FU* fournit la séquence fonctionnelle suivante:

- L'INITIATOR UE sollicite une réponse de type "*I-am-well*" à partir du RESPONDER UE, dans une association établie.
- Le RESPONDER UE répond sur la même association.

5.7.2 Règles de coordination

5.7.2.1 Utilisation d'une association

Toutes les interactions TASE.1 qui font partie d'une invocation de la *Test Association FU* sont véhiculées par une association unique. Les associations sont permanentes ou dynamiques et doivent avoir les caractéristiques spécifiées en 5.7.2.3.1.

5.7.2.2 Relation avec les autres FU

5.7.2.2.1 Invoking FUs

La *Test Association FU* n'est invoquée par aucune autre FU.

5.7.2.2.2 Invoked FUs

La *Test Association FU* n'invoque aucune autre FU.

³⁰⁾ Aucune vérification du paramètre Acceptor n'est spécifiée, afin d'adapter les transformations d'adresses de réseau: le RESPONDER est ainsi autorisé à répondre en utilisant une adresse de réseau autre que l'adresse à laquelle il est connu dans l'INITIATOR.

Errors in the *TASEBindConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Initiator >< Initiator in corresponding <i>TASEBindReq</i> primitive, and Result = <i>result-ok</i> , ³⁰⁾ (INITIATOR FU identity mix-up in RESPONDER.)	Report the error to the Coordinating Function, and trigger local abrupt termination of the association.
Zero octets in User-data, and Result = <i>result-ok</i> .	Treat as normal User-data. NOT an error; enter running state as normal.
Result >< <i>result-ok</i>	Report the error to the local Coordinating Function, and enter the idle state.

Errors in the *TASEUnbindConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Result >< <i>result-ok</i>	Report the error to the Coordinating Function, and trigger local abrupt termination of the association.

5.7 Test Association FU

Type: Primary.

5.7.1 Function

A *Test Association FU* invocation provides the following functional sequence:

- The INITIATOR UE solicits an "*I-am-well*" type answer from the RESPONDER UE, on an established association.
- The RESPONDER UE answers, on the same association.

5.7.2 Coordination rules

5.7.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Test Association FU* are conveyed by one single association. The association may be permanent or dynamic, and shall have the characteristics as specified in 5.7.2.3.1.

5.7.2.2 Relation to other FUs

5.7.2.2.1 Invoking FUs

The *Test Association FU* is not invoked by any other FU.

5.7.2.2.2 Invoked FUs

The *Test Association FU* does not invoke any other FU.

³⁰⁾ No checking of the Acceptor parameter is specified, in order to accommodate network address transformations: the RESPONDER is thus allowed to respond with a network address other than the address by which it is known to the INITIATOR.

5.7.2.2.3 Disrupting FUs

Il est permis à la *Permanent Association FU* (association permanente) ou à la *Dynamic Association FU* (association dynamique) d'interrompre la *Test Association FU*, après réception de la primitive *TASEAbortInd* pour l'association concernée.

5.7.2.2.4 Disrupted FUs

La *Test Association FU* n'interrompt aucune autre FU.

5.7.2.3 Invocation

5.7.2.3.1 Conditions préalables

Les FU suivantes doivent avoir été invoquées avant toute invocation de la *Test Association FU*:

- La *Permanent Association FU* ou la *Dynamic Association FU*

L'invocation de la *Permanent Association FU* (*Dynamic Association FU*) doit toujours être en cours de fonctionnement lors de l'invocation de la *Test Association FU*.

La *Permanent Association FU* (*Dynamic Association FU*) doit avoir été invoquée pour créer et conserver l'association à utiliser pour les interactions liées à l'invocation en cours de la *Test Association FU*. Il est permis à la *Test Association FU* d'être invoquée dans n'importe quelle association, à l'initiative de l'INITIATOR UE. Cependant, si une association est à établir uniquement pour invoquer la *Test Association FU*, cette association doit avoir la paire A-suffix, comme spécifié dans le tableau de 5.2.1.

5.7.2.3.2 Restrictions

Les associations simultanées multiples de la *Test Association FU* sur une association unique ne sont pas autorisées.

5.7.2.3.3 Invoking events

Il est admis d'invoquer la partie INITIATOR de la *Test Association FU* par:

- Une requête locale par le biais de la Coordinating Function, dont la source ne fait pas partie du domaine d'application du présent rapport technique³¹⁾.

L'invocation de la partie RESPONDER de la *Test Association FU* est tentée chaque fois qu'une primitive *TestConnectionInd* est reçue par le biais d'une association, quelles que soient les caractéristiques de l'association.

5.7.2.4 Fin

5.7.2.4.1 Orderly termination

La partie INITIATOR d'une invocation de la *Test Association FU* se termine toujours de façon ordonnée, à la réception de la primitive *TestConnectionConf* attendue.

La partie RESPONDER d'une invocation de la *Test Association FU* se termine de façon ordonnée après l'émission de la primitive *TestConnectionResp*.

Il est également permis de déclencher une fin de la *Test Association FU* au moyen d'une erreur d'encombrement. Voir 5.7.3.2.5.

³¹⁾ De telles invocations sont soit pilotées par horloge, soit déclenchées par l'opérateur.

5.7.2.2.3 Disrupting FUs

The *Test Association FU* may be disrupted by the *Permanent Association FU* (permanent association) or the *Dynamic Association FU* (dynamic association), subsequent to reception of the *TASEAbortInd* primitive for the association concerned.

5.7.2.2.4 Disrupted FUs

The *Test Association FU* does not disrupt any other FU.

5.7.2.3 Invocation

5.7.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Test Association FU*:

- The *Permanent Association FU* or the *Dynamic Association FU*

The *Permanent Association FU* (*Dynamic Association FU*) invocation shall still be running at the time of invocation of the *Test Association FU*.

The *Permanent Association FU* (*Dynamic Association FU*) shall have been invoked in order to create and maintain the association to be used for the interactions related to the current invocation of the *Test Association FU*. The *Test Association FU* may be invoked on any association, at the INITIATOR UE's discretion. However, if an association is to be established solely for the purpose of invoking the *Test Association FU*, this association shall have the A-suffix pair as specified in the table in 5.2.1.

5.7.2.3.2 Restrictions

Multiple simultaneous invocations of the *Test Association FU* on a single association are not allowed.

5.7.2.3.3 Invoking events

The INITIATOR part of the *Test Association FU* may be invoked by:

- Local request via the Coordinating Function, the original source of which is outside the scope of this technical report³¹⁾.

Invocation of the RESPONDER part of the *Test Association FU* is attempted whenever a *TestConnectionInd* primitive is received via an association, regardless of the characteristics of the association.

5.7.2.4 Termination

5.7.2.4.1 Orderly termination

The INITIATOR part of a *Test Association FU* invocation always terminates itself in an orderly manner upon reception of the expected *TestConnectionConf* primitive.

The RESPONDER part of a *Test Association FU* invocation terminates itself in an orderly manner after issuing the *TestConnectionResp* primitive.

Such termination of the *Test Association FU* may also be triggered by congestion error. See 5.7.3.2.5.

³¹⁾ Such invocations typically may be timer-driven or operator-triggered.

5.7.2.4.2 Disruption

Il est permis d'interrompre la partie INITIATOR et la partie RESPONDER de l'invocation de la *Test Association FU* par:

- La rupture d'une autre invocation de FU. Voir 5.7.2.2.3.

5.7.3 Procédures

5.7.3.1 Primitives de service TASE.1

Le service TASE.1 élémentaire suivant est utilisé par la *Test Association FU*:

Test-Connection

5.7.3.1.1 Sequence

INITIATOR UE	TASE.1	RESPONDER UE
<i>TestConnectionReq</i>	----->	<i>TestConnectionInd</i>
<i>TestConnectionConf</i>	<-----	<i>TestConnectionResp</i>

5.7.3.1.2 Parameter values

Test-Connection:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Result	(Non applicable)	Toujours égal à la valeur <i>result-ok</i>

5.7.3.2 Traitement des erreurs

5.7.3.2.1 FU disruption

Rupture par la *Permanent Association FU* (*Dynamic Association FU*):

La rupture de la partie INITIATOR et de la partie RESPONDER de l'invocation en cours de la *Test Association FU* doit être déclenchée localement, comme faisant partie de la procédure de rupture de la partie correspondante de la *Permanent Association FU* (*Dynamic Association FU*).

Les deux parties de l'invocation en cours de la *Test Association FU* doivent se terminer correctement, aucune des deux parties ne devant tenter d'émettre une primitive associée à la fin elle-même³²⁾.

5.7.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Test Association FU* n'est pas présente sur l'INITIATOR UE:

Les requêtes d'invocation sont toujours générées localement, voir 5.7.2.3. Le traitement de ce type d'erreur se fait donc au niveau local et ne fait pas partie du domaine d'application de ce rapport.

³²⁾ Les procédures de nettoyage local ne sont pas spécifiées par le présent rapport technique.

5.7.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Test Association FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 5.7.2.2.3.

5.7.3 Procedures

5.7.3.1 TASE.1 service primitives

The following elementary TASE.1 service is used by the *Test Association FU*:

Test-Connection

5.7.3.1.1 Sequence

INITIATOR UE	TASE.1	RESPONDER UE
<i>TestConnectionReq</i>	----->	<i>TestConnectionInd</i>
<i>TestConnectionConf</i>	<-----	<i>TestConnectionResp</i>

5.7.3.1.2 Parameter values

Test-Connection:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Result	(Not applicable)	Always equal to value <i>result-ok</i>

5.7.3.2 Error handling

5.7.3.2.1 FU disruption

Disruption by the *Permanent Association FU* (*Dynamic Association FU*):

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Test Association FU* shall be triggered locally, as a part of the disruption procedure for the corresponding part of the *Permanent Association FU* (*Dynamic Association FU*).

Both parts of the current invocation of the *Test Association FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself³²⁾.

5.7.3.2.2 Illegal invocation attempt

FU not present:

If the *Test Association FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see 5.7.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this report.

³²⁾ Local clean-up procedures are not specified by this technical report.

Si la *Test Association FU* n'est pas présente sur un RESPONDER UE:

Le RESPONDER User Element doit ignorer la primitive *TestConnectionInd* entrante.

FU présente, mais tentative illégale:

Les tentatives d'invocations multiples simultanées de la *Test Association FU* sur une association donnée dans un INITIATOR UE, sont traitées localement et ne font pas partie du domaine d'application du présent rapport technique.

Les tentatives d'invocations multiples simultanées de l'*Association FU* d'une association donnée sur un RESPONDER UE sont exclues par des mécanismes générés dans le TASE.1.

5.7.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>TestConnectionConf</i>
FU pas en fonctionnement	Ignorer, ou indication d'erreur locale
FU en fonctionnement, inactive	Ignorer, ou indication d'erreur locale
FU en fonctionnement, en attente d'une réponse de la part du RESPONDER	(Normal)

Dans la partie RESPONDER, une primitive *TestConnectionInd* entrante n'est jamais considérée comme hors contexte.

5.7.3.2.4 Timing errors

Erreur dans la partie INITIATOR:

Aucune erreur d'horloge ne peut se produire dans la partie INITIATOR.

Erreur dans la partie RESPONDER:

Erreur	Réaction de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à la <i>TestConnectionInd</i>	Sur le RESPONDER: <i>TASEAbortInd</i>, avec Reason = <i>misbehaviour-of-local-service-user</i>. Erreur locale lors d'une tentative d'émission de <i>TestConnectionResp</i> Sur l'INITIATOR: <i>TestConnectionConf</i>, avec Result = <i>remote-service-user-unavailable</i>.	Partie RESPONDER: Ignorer, ou indication d'erreur locale Partie INITIATOR: Voir 5.7.3.2.6.

5.7.3.2.5 Congestion error

Sur la partie INITIATOR de la FU, intervenant avec une tentative de *TestConnectionReq*:

- terminer l'invocation de FU localement. Voir 4.2.1.

If the *Test Association FU* is not present in an RESPONDER UE:

The RESPONDER User Element shall ignore the incoming *TestConnectionInd* primitive altogether.

FU present, but attempt illegal:

Attempts of multiple simultaneous invocations of the *Test Association FU* on a given association in an INITIATOR UE is a local issue, outside the scope of this technical report.

Attempts of multiple simultaneous invocations of the *Test Association FU* on a given association in a RESPONDER UE are precluded by mechanisms within the TASE.1.

5.7.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>TestConnectionConf</i>
FU not running	Ignore, or local error indication
FU running, idle	Ignore, or local error indication
FU running, waiting for answer from RESPONDER	(Normal)

In the RESPONDER part, an incoming *TestConnectionInd* primitive is never considered out of context.

5.7.3.2.4 Timing errors

Error in INITIATOR part:

There can occur no timing errors in the INITIATOR part.

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>TestConnectionInd</i>	In RESPONDER: <i>TASEAbortInd</i>, with Reason = <i>misbehaviour-of-local-service-user</i>. Local error from eventual attempt at issuing <i>TestConnectionResp</i> In INITIATOR: <i>TestConnectionConf</i>, with Result = <i>remote-service-user-unavailable</i>.	RESPONDER part: Ignore, or local error indication INITIATOR part: See 5.7.3.2.6.

5.7.3.2.5 Congestion error

In INITIATOR part of the FU, occurring with a *TestConnectionReq* attempt:

- Terminate FU invocation locally. See 4.2.1.

Sur la partie RESPONDER de la FU, intervenant avec une tentative de *TestConnectionResp*:

- terminer l'invocation de FU localement, normalement, sauf, si nécessaire, pour le rapport d'erreurs locales. (La partie INITIATOR de la FU sera terminée après l'expiration du délai.)

5.7.3.2.6 TASE.1 service primitive parameter errors

Erreurs dans la primitive *TestConnectionConf* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Result >< <i>result-ok</i>	Pas d'action spécifique autre que l'enregistrement de « <i>Result not ok</i> ».

6 Function group identification de données

6.1 Attributs de groupe

Un groupe est un ensemble logique d'objets donnés possédant des caractéristiques similaires. Un objet peut appartenir à plusieurs groupes. Un groupe valide est autorisé à contenir de 1 ³³⁾ à 255 objets; voir 6.1.3.

Les groupes peuvent être créés et/ou modifiés dans un système RESPONDER de deux manières différentes:

- a) Par le biais de l'interface TASE.1, comme spécifié par ce rapport technique.
- b) Par tout moyen ne faisant pas appel à l'interface TASE.1 et n'étant pas indiqué dans ce rapport (groupes fixes prédéfinis).

Aucune FU ne doit, pendant qu'elle gère la configuration d'un groupe ou ses données associées, faire une distinction entre les deux origines possibles de la configuration (TASE ou non TASE). Toutefois, une distinction fondée sur les valeurs des attributs indiquant l'origine est à la fois autorisée et obligatoire; se reporter ci-dessous à l'attribut Persistent.

Tout groupe, quelle que soit son origine, est caractérisé par le jeu d'attributs suivants, qui sont décrits dans les sections ci-dessous:

- Group type;
- Group number;
- Group size;
- Object identifier size;
- Persistent;
- Static;
- Priority;
- Object set:

Pour chaque objet du groupe, la paire d'attributs:

- Object number;
- Object identifier.

³³⁾ Un groupe contenant 0 objet est créé mais non défini, et n'est pas considéré comme valide ici.

In RESPONDER part of the FU, occurring with a *TestConnectionResp* attempt:

- Terminate FU invocation locally, as normal except for local error report if desired.
(INITIATOR part of FU will be terminated after timeout.)

5.7.3.2.6 TASE.1 service primitive parameter errors

Errors in the *TestConnectionConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Result >< <i>result-ok</i>	No special action other than registering " <i>Result not ok</i> ".

6 The Data Identification Function Group

6.1 Group attributes

A group is a logical collection of data objects of similar characteristics. One object may be a member of more than one group. A valid group may contain 1³³⁾ – 255 objects; see 6.1.3.

Groups may be created and/or modified in a RESPONDER system in two different ways:

- a) Via the TASE.1 interface, as specified by this technical report.
- b) By any means not involving the TASE.1 interface, not specified by this technical report (fixed predefined groups).

No FU shall during its handling of a group configuration or its associated data discriminate between the two possible configuration origins (TASE/non-TASE), as such. Discrimination based on attribute values reflecting the origin is both allowed and mandatory, however; see attribute Persistent, below.

Any group, irrespective of configuration origin, is characterised by the following set of attributes, described in subsequent sections:

- Group type;
- Group number;
- Group size;
- Object identifier size;
- Persistent;
- Static;
- Priority;
- Object set:

For each object in the group, the component attribute pair:

- Object number;
- Object identifier.

³³⁾ A group containing 0 object is created but not defined, and is not considered valid here.

Les attributs suivants sont définis dans le CS(R) par la *Group Management FU* invoquée par la fonction *G-Create*:

- Group type;
- Group number;
- Group size;
- Object identifier size;
- Persistent;
- Static;
- Priority.

L'attribut suivant est défini dans le CS(R) par la *Group Definition FU*:

- Object set (certains ou tous les objets).

Seul les attributs suivants³⁴⁾ doivent être modifiés dans le CS(R) par la *Group Management FU* invoquée par la fonction *G-Change*:

- Persistent (seule la transition false/true est légale);
- Static;
- Priority.

6.1.1 Group type

Le tableau ci-dessous énumère les types de groupes normalisés par la spécification TASE.1³⁵⁾ Chaque type est décrit brièvement et les FU du *FUNCTION GROUP TRANSFERT DE DONNEES* qu'il est admis d'invoquer pour ce type de groupe sont indiquées.

Les noms des types de groupes n'impliquent aucune restriction quant à leur utilisation.

Group type	Description	FU de transfert de données
<i>Measurand-group</i>	Valeurs à virgule flottante	DRFU, DPRFU, DUPFU, DUSFU
<i>Status-group</i>	Valeurs à trois états (on/off/between)	DRFU, DPRFU, DUPFU, DUSFU
<i>Discrete-group</i>	Entiers à deux compléments	DRFU, DPRFU, DUPFU, DUSFU
<i>Logical-breaker-status-group</i>	Valeurs composites à trois états	DRFU, DPRFU, DUPFU, DUSFU
<i>Binary-command-group</i>	Commandes à deux états (on/off)	DSFU
<i>Analog-setpoint-group</i>	Valeurs de commande à virgule flottante	DSFU
<i>Digital-setpoint-group</i>	Valeurs de commandes entières à deux compléments	DSFU
<i>Text-message-group</i>	Chaîne de caractères ASCII générale	DRFU, DPRFU, DUPFU, DUSFU

Des informations complémentaires se trouvent dans la description de chaque FU et dans la documentation TASE.1 correspondante (CEI 60870-6-501 et CEI 60870-6-502).

³⁴⁾ Voir la CEI 60870-6-501, 9.4.

³⁵⁾ Des types de groupes supplémentaires, définis pour des conventions régionales, sont autorisés.

The following attributes are set in the CS(R) by the *Group Management FU* invoked with function *G-Create*:

- Group type;
- Group number;
- Group size;
- Object identifier size;
- Persistent;
- Static;
- Priority.

The following attribute is set in the CS(R) by the *Group Definition FU*:

- Object set (some or all objects)

Only the following³⁴⁾ attributes shall be modified in the CS(R) by the *Group Management FU* invoked with function *G-Change*:

- Persistent (only transition from *false* to *true* is legal);
- Static;
- Priority.

6.1.1 Group type

The table below lists the group types that are standardised by the TASE.1 specification³⁵⁾. A short description is given for each type, and the FUs within the DATA TRANSFER FUNCTION GROUP that may be invoked for that group type are identified.

Group type names do not imply any restrictions on the usage.

Group type	Description	DATA TRANSFER FUs
<i>Measurand-group</i>	Floating-point values	DRFU, DPRFU, DUPFU, DUSFU
<i>Status-group</i>	Three-state values (on/off/between)	DRFU, DPRFU, DUPFU, DUSFU
<i>Discrete-group</i>	Two-complement integers	DRFU, DPRFU, DUPFU, DUSFU
<i>Logical-breaker-status-group</i>	Composite three-state values	DRFU, DPRFU, DUPFU, DUSFU
<i>Binary-command-group</i>	Two-state commands (on/off)	DSFU
<i>Analog-setpoint-group</i>	Floating-point command values	DSFU
<i>Digital-setpoint-group</i>	Two-complement integer command values	DSFU
<i>Text-message-group</i>	General ASCII character string	DRFU, DPRFU, DUPFU, DUSFU

Further information may be found in the individual FU descriptions and in relevant TASE.1 reportation (IEC 60870-6-501 and IEC 60870-6-502).

³⁴⁾ See IEC 60870-6-501, 9.4.

³⁵⁾ Additional group types, defined for regional conventions, are allowed.

6.1.2 Group number

Un nombre entier est associé à chaque groupe pour identification. Il est à utiliser avec la TAPI³⁶⁾. Ce Group number doit être unique dans le cadre du Configuration Set auquel il appartient. En conséquence, dans tout système INITIATOR ou RESPONDER, le Group number à lui seul permet d'identifier le groupe.

Les valeurs autorisées pour les Group numbers vont de 1 à 32767. Noter que le Group number 0 n'est pas autorisé³⁷⁾.

Les Group numbers sont proposés par l'INITIATOR UE au RESPONDER UE, dans le cadre d'une invocation de la Group Management UE pour ce numéro, par la fonction *G-Create*.

Le RESPONDER UE est autorisé à accepter ou rejeter un numéro proposé en fonction des conditions locales³⁸⁾. Tout RESPONDER UE doit cependant supporter au minimum la plage de Group numbers allant de 1 à 30.

Du fait que les différents Configuration Sets partagés par tout système INITIATOR ou RESPONDER doivent être définis indépendamment, chaque système RESPONDER est tenu de supporter l'utilisation simultanée avec un nombre quelconque de partenaires d'un numéro de groupe attribué par le système INITIATOR homologue. Le nombre total de groupes d'un UE n'est limité que par des raisons liées à l'implémentation.

Les Group numbers ne doivent pas être associés logiquement aux identificateurs d'association.

6.1.3 Group size

La Group size est le nombre maximal d'objets que l'attribut Object set du groupe est autorisé à contenir. La plage de valeurs légale va de 1 à 255.

Les Group sizes sont proposées par l'INITIATOR UE au RESPONDER UE de la même manière que les Group numbers (ci-dessus). Le RESPONDER UE n'est autorisé à supporter qu'un sous-ensemble de la plage de valeurs légale. De ce fait, le traitement des défaillances de la *Group Management FU (G-Create)* n'est pas spécifiée par ce rapport. Tout RESPONDER UE doit supporter au minimum la valeur suivante de Group size: 20.

La Group size est indiquée par le paramètre MaxObj.

6.1.4 Object identifier size

L'Object identifier size est le nombre maximal d'octets que tout Object Identifier de l'attribut Object set du groupe est autorisé à contenir, à l'exclusion de l'octet indicateur de longueur. La plage légale va de 1 à 255.

³⁶⁾ Pour une utilisation locale, il est admis d'identifier les groupes TASE.1 d'une autre façon. Toutefois, pour la TAPI, l'identificateur est toujours le numéro du groupe.

³⁷⁾ Dans le *MixedDataReq* PDU, par exemple, le group number 0 termine le champ de données.

³⁸⁾ Il se peut, par exemple, qu'il existe des inconsistences permanentes entre le CS(I) et le CS(R) concernant des groupes fixes prédéfinis: il se peut qu'un numéro de groupe soit libre du point de vue de l'INITIATOR UE (dans le CS(I)), mais qu'il soit occupé par un groupe fixe prédéfini dans le CS(R). Il se peut également que le RESPONDER UE ne soit pas conçu pour supporter la totalité des numéros de groupes autorisés. Dans ces deux cas, il est nécessaire que le RESPONDER UE rejette le numéro de groupe proposé. Les mécanismes de récupération adaptés à de telles situations ne sont pas spécifiés dans le présent rapport technique.

6.1.2 Group number

Each group has an identifying associated integer, for use with the TAPI³⁶⁾. This Group number shall be unique within the scope of the Configuration Set to which it belongs. Consequently, in any INITIATOR or RESPONDER system, the Group number is by itself sufficient information to identify the group.

Legal value for Group numbers are 1 – 32767. Note that Group number 0 is not allowed³⁷⁾.

Group numbers are proposed by the INITIATOR UE to the RESPONDER UE, within a *Group Management FU* invocation for that number, with function *G-Create*.

The RESPONDER UE may accept or reject a proposed number, according to local conditions³⁸⁾. As a minimum, however, any RESPONDER UE shall support the following range of Group numbers: 1 – 30.

Because the different Configuration Sets that any INITIATOR or RESPONDER system shares shall be independently defined, any RESPONDER system is required to support use of a Group number given by the peer INITIATOR system with any number of partners simultaneously. The total number of groups in a UE is not restricted by other than implementation dependent reasons.

Group numbers shall not be logically coupled with association identifiers.

6.1.3 Group size

Group size is the maximum number of objects that the attribute Object set of the group may contain. The legal value range is 1 – 255.

Group sizes are proposed by the INITIATOR UE to the RESPONDER UE in a manner similar to Group numbers (above). The RESPONDER UE may support only a subset of the legal value range. Handling of *Group Management FU (G-Create)* failures because of this, is not specified by this report. As a minimum, any RESPONDER UE shall support the following value of Group size: 20.

Group size is denoted by the parameter MaxObj.

6.1.4 Object identifier size

Object identifier size is the maximum number of octets that any Object identifier within the attribute Object set of the group may contain, excluding the length indicator octet. The legal range is 1 – 255.

³⁶⁾ For local use, TASE.1 groups may well be identified by other means. However, at the TAPI, the identifying unit is always this group number.

³⁷⁾ In the *MixedDataReq* PDU, for example, group number 0 terminates the data field.

³⁸⁾ For example, there may exist permanent inconsistencies between the CS(I) and CS(R), concerning fixed, predefined groups: a group number may be free from the INITIATOR UE's point of view (in the CS(I)), but occupied by a fixed, predefined group in the CS(R). Also, the RESPONDER UE may not be designed to support the full range of legal group numbers. The RESPONDER UE will have to reject the proposed group number in both these cases. Recovery mechanisms for such situations are not specified by this technical report.

Les Object identifier sizes sont proposées par l'INITIATOR UE au RESPONDER UE de la même manière que les Group numbers (ci-dessus). Le RESPONDER UE n'est autorisé à supporter qu'un sous-ensemble de la plage de valeurs légale. De ce fait, le traitement des défaillances de la *Group Management FU (G-Create)* n'est pas spécifié par ce rapport. Tout RESPONDER UE doit supporter au minimum la valeur suivante d'Object identifier size: 12.

Object identifier size est défini par le paramètre MaxObjSize.

6.1.5 Persistent

Booléen possédant la signification suivante:

true: Le groupe ne peut être ni supprimé/modifié ni redéfini par le biais de l'interface TASE.1. Toute invocation de la *Group Management FU* ou de la *Group definition FU* doit échouer pour ces groupes³⁹⁾.

Ces groupes sont qualifiés de permanents et, s'ils ne sont pas non plus d'origine TASE.1, de prédéfinis.

false: Il est autorisé de jouer à volonté sur la configuration du groupe par le biais de l'interface TASE.1.

Ces groupes sont qualifiés de non-permanents ou de dynamiques.

Noter que s'il est souhaitable d'exclure une configuration de groupe d'origine non TASE.1 par une manipulation par le biais de l'interface TASE.1, il suffit de faire en sorte que l'attribut Persistent=*true* pour ce groupe dans le CS(R).

Noter également qu'il est autorisé de faire passer Persistent de *false* à *true* dans le CS(R) avec la *Group Management FU*. Cela transforme tout groupe dynamique en groupe permanent. Il est par contre interdit d'exécuter une transformation en sens inverse: un groupe permanent doit toujours être modifié ou supprimé, si nécessaire, par des moyens non TASE.1.

6.1.6 Static

Booléen possédant la signification suivante:

true: Le jeu d'objets du groupe ne peut être ni défini, ni redéfini par une invocation de la *Group Definition FU* destinée au groupe, sauf dans le cas où le groupe ne contient aucun Object identifier dans le CS(R), qui doit être traité par le RESPONDER UE comme si Static était égal à *false*⁴⁰⁾.

false: Le jeu d'objets du groupe peut être défini ou redéfini à volonté par des invocations de la *Group Definition FU* destinées au groupe.

La valeur de Persistent doit toujours prévaloir sur celle de Static, n'autorisant une (re)définition que si Persistent=*false*.

6.1.7 Priority

Priority est un entier compris entre 0 et 15, qui régit:

a) Pour le Spontaneous Data Transfer:

Le mécanisme de commande de la file de données en sortie pour toutes les *Unsolicited Spontaneous Data Transfer FU* et invocations dans un système RESPONDER donné: les données appartenant à un groupe ayant une valeur de Priority donnée doivent obtenir la priorité sur toutes les données des groupes ayant des valeurs de Priority inférieures. Voir 7.5 pour plus de détails.

³⁹⁾ Les règles de manipulation par des moyens autres que TASE.1 sortent du domaine d'application du présent rapport technique.

⁴⁰⁾ L'objet de cette exception est de rendre possible la reconfiguration (par l'invocation de la *Restart Reconfigure FU*) des groupes pour lesquels Static=*true*.

Object identifier sizes are proposed by the INITIATOR UE to the RESPONDER UE in a manner similar to Group numbers (above). The RESPONDER UE may support only a subset of the legal value range. Handling of *Group Management FU (G-Create)* failures because of this, is not specified by this report. As a minimum, any RESPONDER UE shall support the following value of Object identifier size: 12.

Object identifier size is denoted by the parameter MaxObjSize.

6.1.5 Persistent

A boolean value, with the following semantics:

- true*: The group can be neither deleted/changed nor redefined via the TASE.1 interface. Any invocation of the *Group Management FU* or the *Group Definition FU* shall fail for such groups³⁹⁾.
Such groups are termed permanent, and if also of non-TASE.1 origin, predefined.
- false*: The group configuration may be manipulated at will via the TASE.1 interface.
Such groups are termed non-permanent, or dynamic.

Note that if it is desirable to exclude a group configuration of non-TASE.1 origin from any manipulative access via the TASE.1 interface, it suffices to set the attribute Persistent = *true* for that group in the CS(R).

Note also that changing Persistent from *false* to *true* in the CS(R) with the *Group Management FU* is allowed, effectively transforming any dynamic group into a permanent one. Transformation in the opposite direction, on the other hand, is illegal: a permanent group must always be changed or removed by non-TASE.1 means, if necessary.

6.1.6 Static

A boolean value, with the following semantics:

- true*: The object set of the group can be neither defined nor redefined with a *Group Definition FU* invocation for the group, except for the case of the group containing no Object identifier in the CS(R), which shall be treated as if Static were equal to *false*, by the RESPONDER UE⁴⁰⁾.
- false*: The object set of the group may be defined or redefined at will, with *Group Definition FU* invocations for the group.

The value of Persistent shall always override the value of Static, allowing (re)definition only when Persistent = *false*.

6.1.7 Priority

Priority is an integer value within the range 0 – 15, governing:

a) For Spontaneous Data Transfer:

The mechanism for ordering the output data queue across all *Unsolicited Spontaneous Data Transfer FUs* and invocations in a given RESPONDER system: Data belonging to a group with a given Priority value shall be given precedence over all data belonging to groups with lower Priority values. See 7.5 for details.

³⁹⁾ Rules for manipulation by non-TASE.1 means are not within the scope of this technical report.

⁴⁰⁾ The purpose of this exception is to enable reconfiguration (invocation of *Restart Reconfigure FU*) of groups for which Static = *true*.

b) Pour le Periodic Data Transfer:

Les longueurs de périodes du RESPONDER UE pour les diverses *invocations Unsolicited Periodic Data Transfer FU*, indirectement. Voir 7.4.

Priority régit pour un groupe donné soit la file de données de sortie, soit la longueur de la période: cela est déterminé exclusivement par la FU en fonctionnement pour le groupe. Cette distinction ne fait pas partie en tant que telle de la configuration du groupe.

Les valeurs de Priority sont proposées par l'INITIATOR UE au RESPONDER UE d'une manière similaire aux Group numbers (voir ci-dessus).

Un RESPONDER UE est autorisé ou non à supporter ce mécanisme de priorité⁴¹⁾. Les RESPONDER UE qui supportent le mécanisme doivent accepter toutes les valeurs de Priority de la plage légale⁴²⁾. Les RESPONDER UE qui ne supportent pas le mécanisme ne doivent accepter que la valeur 0. Le traitement des défaillances des *Group Management FU (G-Create/G-Change)* dues à un RESPONDER UE ne supportant pas le mécanisme n'est pas spécifié dans le présent rapport technique.

6.1.8 Object set

L'Object set d'un groupe est l'ensemble des identificateurs de valeurs des données symboliques, ou Object identifiants, pour ce groupe. A l'intérieur du groupe, chaque Object identifier se voit assigner un numéro de référence appelé Object number. Il est admis qu'un Object identifier donné appartienne à un nombre quelconque de groupes et qu'il ait un Object number différent dans ces différents groupes.

La *Group Definition FU* est le moyen par lequel un INITIATOR UE demande au système RESPONDER de mettre les Object sets des groupes (re)définissables du CS(R) en conformité avec le CS(I) du système INITIATOR.

La correspondance exacte entre le CS(I) et le CS(R) pour les groupes fixes prédéfinis doit être assurée par des moyens non TASE.1, qui ne font pas partie du domaine d'application du présent rapport technique.

6.1.8.1 Object number

Numéro de référence d'un Object identifier dans un groupe donné. La plage autorisée est la suivante: 1 – <valeur de Group size>. Noter que Object number 0 n'est pas autorisé.

Les Object numbers sont assignés implicitement par le RESPONDER UE à la définition du groupe du CS(R), dans l'ordre selon lequel les divers Object identifiants apparaissent dans le paramètre Objid de la primitive de service *DefineGroupInd*; voir 6.3. La valeur du paramètre Index1 de la primitive détermine directement l'Object number du premier Object identifier du paramètre Objid. L'Object identifier suivant du paramètre Objid se voit assigner l'Object number Index1+1, le suivant Index 1+2 et ainsi de suite, jusqu'à ce que le paramètre Objid ait été épuisé ou que l'Object number Index2 ait été assigné.

⁴¹⁾ Le support de ce mécanisme signifie le support de l'ordonnancement de la file de sortie et la détermination de la longueur de période.

⁴²⁾ Une granularité des décisions effectives inférieure à 1/15 est toutefois autorisée: il est admis de définir des sous-plages locales avec une valeur effective de Priority égale, mais il convient d'obtenir au moins deux valeurs effectives.

b) For Periodic Data Transfer:

The period lengths in the RESPONDER UE for the individual *Unsolicited Periodic Data Transfer FU* invocations, in an indirect manner. See 7.4.

Whether Priority will have the effect of governing output data queue or period length on behalf of a given group is determined exclusively by which FU is running for the group. The distinction is not a part of the group configuration, as such.

Priority values are proposed by the INITIATOR UE to the RESPONDER UE in a manner similar to Group numbers (above).

A RESPONDER UE may or may not support this priority mechanism⁴¹⁾. RESPONDER UEs that support the mechanism shall accept all Priority values within the legal range⁴²⁾. RESPONDER UEs that do not support the mechanism, shall accept only the value 0. Handling of *Group Management FU (G-Create/G-Change)* failures because of a RESPONDER UE not supporting the mechanism, is not specified by this technical report.

6.1.8 Object set

The Object set of a group is the set of symbolic data value identifiers, or Object identifiers, for the group. Within the group, each Object identifier is given a reference number: the Object number. A given Object identifier may be a member of any number of groups, and may have different Object number in different groups.

The *Group Definition FU* is the means by which an INITIATOR UE instructs the RESPONDER system to bring the Object sets of (re)definable groups at the CS(R) into accordance with the CS(I) at the INITIATOR system.

Exact match between the CS(I) and the CS(R) for fixed, predefined groups must be ensured by separate non-TASE.1 means, not within the scope of this technical report.

6.1.8.1 Object number

The reference number of an Object identifier within a given group. The legal range is 1 – <value of Group size>. Note that Object number 0 is not legal.

Object numbers are assigned to the group definition in the CS(R) by the RESPONDER UE implicitly, by the order in which the individual Object identifiers appear in the Objid parameter of the *Define Group Ind* service primitive; see 6.3. The value of the Index1 parameter of the primitive directly determines the Object number of the first Object identifier in the Objid parameter. The next Object identifier in the Objid parameter is given Object number Index1+1, the next thereafter is given Index1+2, and so on, until either the Objid parameter is exhausted or Object number Index2 has been assigned.

⁴¹⁾ Support of this mechanism shall mean support of both output queue ordering and period length determination.

⁴²⁾ However, the granularity of actual decisions may be less than 1/15: local subranges with equal effective value of Priority may be defined, but at least two effective values shall be the result.

6.1.8.2 Object identifier

L'Object identifier sert de lien logique entre l'environnement TASE.1 et l'environnement local de la base de données/acquisition de données dans un système INITIATOR/RESPONDER: le système INITIATOR/RESPONDER doit utiliser l'Object identifier afin d'identifier⁴³⁾ la valeur de donnée locale à associer logiquement à une combinaison donnée de Group number et d'Object number. Une spécification plus approfondie des mécanismes d'interface base de données/acquisition de données est en dehors du domaine d'application du présent rapport technique.

Les Object identifiers sont copiés par le RESPONDER UE à partir du paramètre Objid de la primitive de service *DefineGroupInd* dans le CS(R); voir 6.3.

6.2 Group Management FU

Type: Primary.

6.2.1 Fonction

Une invocation de la *Group Management FU* autorise l'INITIATOR UE qui effectue l'invocation à modifier les attributs de groupe dans le CS(R) d'un RESPONDER UE avec l'effet suivant, selon la fonction *Group Management FU* spécifiée:

Fonction	Effet
<i>G-Create</i>	Crée un nouveau groupe: assigne un nouveau jeu d'attributs ayant les valeurs spécifiées
<i>G-Change</i>	Modifie les valeurs d'attributs existantes pour un groupe
<i>G-Delete</i>	Supprime un seul groupe: libère le jeu d'attributs correspondant
<i>Delete-all-groups</i>	Supprime TOUS les groupes: libère le jeu d'attributs pour tous les groupes du CS(R)

6.2.2 Règles de coordination

6.2.2.1 Utilisation d'une association

Toutes les interactions TASE.1 faisant partie d'une invocation de la *Group Management FU* sont véhiculées par une seule association. L'association doit posséder les caractéristiques spécifiées en 6.2.2.3.1.

6.2.2.2 Relations avec les autres FU

6.2.2.2.1 Invoking FUs

La *Group Management FU* peut être invoquée par:

- la *Group Configuration FU*;
- la *Restart Reconfigure FU*.

6.2.2.2.2 Invoked FUs

La *Group Management FU* ne doit invoquer aucune autre FU.

⁴³⁾ Dans un système ou un site donné, il est admis d'identifier les objets par d'autres mécanismes, avec une référence locale à l'Object identifier TASE.1 correspondant.

6.1.8.2 Object identifier

The Object identifier serves as the logical link between the TASE.1 environment and the local database/data-acquisition environment in an INITIATOR/RESPONDER system: the INITIATOR/RESPONDER system shall use the Object identifier in order to identify⁴³⁾ which local data value is to be logically associated with a given combination of Group number and Object number. Further specification of database/data-acquisition interface mechanisms is outside the scope of this technical report.

Object identifiers are copied by the RESPONDER UE from the Objid parameter of the *DefineGroupInd* service primitive into the CS(R); see 6.3.

6.2 Group Management FU

Type: Primary.

6.2.1 Function

A *Group Management FU* invocation allows the invoking INITIATOR UE to modify group attributes in the CS(R) of a RESPONDER UE, with the following effect, according to specified *Group Management FU* function:

Function	Effect
<i>G-Create</i>	Creates new group: allocates a new set of attributes, with specified values
<i>G-Change</i>	Changes existing attribute values, for a group
<i>G-Delete</i>	Deletes a single group: de-allocates the corresponding set of attributes
<i>Delete-all-groups</i>	Deletes ALL groups: de-allocates the set of attributes for all groups in the CS(R)

6.2.2 Coordination rules

6.2.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Group Management FU* are conveyed by one single association. The association shall have the characteristics as specified in 6.2.2.3.1.

6.2.2.2 Relation to other FUs

6.2.2.2.1 Invoking FUs

The *Group Management FU* may be invoked by:

- *Group Configuration FU*;
- *Restart Reconfigure FU*.

6.2.2.2.2 Invoked FUs

The *Group Management FU* shall not invoke any other FU.

⁴³⁾ Within a given system or site, objects may be identified by other mechanisms, with local reference to the corresponding TASE.1 Object identifier.

6.2.2.2.3 Disrupting FUs

Il est admis d'interrompre la *Group Management FU* par:

- la *Permanent Association FU*;
- la *Dynamic Association FU*.

6.2.2.2.4 Disrupted FUs

Il est admis que la *Group Management FU* interrompe les FU suivantes:

- *Group Configuration FU*;
- *Requested Data Transfer FU*;
- *Periodically Requested Data Transfer FU*;
- *Unsolicited Data Transfer FU*;
- *Unsolicited Periodic Data Transfer FU*;
- *Unsolicited Spontaneous Data Transfer FU*;
- *Supervisory Control Data Transfer FU*.

6.2.2.3 Invocation

6.2.2.3.1 Conditions préalables

L'une⁴⁴⁾ des FU suivantes doit avoir été invoquée avant toute invocation de la *Group Management FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*.

L'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* doit être encore en fonctionnement au moment de l'invocation de la *Group Management FU*. Dans le cas de la *Permanent Association FU*, l'association maintenue par l'invocation doit être en fonctionnement (et non interrompue temporairement) à ce moment-là.

La *Permanent Association FU* ou la *Dynamic Association FU* doivent avoir été invoquées afin de créer (et, pour la *Permanent Association FU*, également de maintenir) l'association à utiliser pour les interactions liées à l'invocation en cours de la *Group Management FU*, avec la paire A-suffix spécifiée par le tableau de 5.2.1.

6.2.2.3.2 Restrictions

Pour toute combinaison donnée de systèmes INITIATOR/RESPONDER, des invocations multiples simultanées de la *Group Management FU* pour quel que groupe que ce soit ne sont pas autorisées.

Une *Group Management FU* ne doit pas être invoquée pendant qu'une *Group Definition FU* est en fonctionnement, pour le groupe concerné.

Une *Group Management FU* ne doit pas être invoquée pendant qu'une invocation de la *Group Management FU* ou de la *Group Definition FU* est en fonctionnement pour un autre groupe sur une autre association avec le même partenaire ⁴⁵⁾

⁴⁴⁾ Sous réserve d'une décision locale dans l'INITIATOR UE.

⁴⁵⁾ L'objet de cette règle est d'assurer l'intégrité du paramètre de contrôle de l'intégrité de la configuration pour le système INITIATOR au niveau du système RESPONDER. Ce paramètre est mis à jour par la *Group Management FU* et la *Group Definition FU*; voir 5.4.

6.2.2.2.3 Disrupting FUs

The *Group Management FU* may be disrupted by:

- *Permanent Association FU*;
- *Dynamic Association FU*.

6.2.2.2.4 Disrupted FUs

The *Group Management FU* may disrupt the following FUs:

- *Group Configuration FU*;
- *Requested Data Transfer FU*;
- *Periodically Requested Data Transfer FU*;
- *Unsolicited Data Transfer FU*;
- *Unsolicited Periodic Data Transfer FU*;
- *Unsolicited Spontaneous Data Transfer FU*;
- *Supervisory Control Data Transfer FU*.

6.2.2.3 Invocation

6.2.2.3.1 Prerequisites

One⁴⁴⁾ of the following FUs shall have been invoked preceding any invocation of the *Group Management FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*.

The *Permanent Association FU* or *Dynamic Association FU* invocation shall still be running at the time of invocation of the *Group Management FU*. In the case of *Permanent Association FU* the association maintained by the invocation shall be running (not temporarily broken) at that time.

The *Permanent Association FU* or *Dynamic Association FU* shall have been invoked in order to create (and, for the *Permanent Association FU*, also to maintain) the association to be used for the interactions related to the current invocation of the *Group Management FU*, with the A-suffix pair specified by the table in 5.2.1.

6.2.2.3.2 Restrictions

For any given INITIATOR/RESPONDER system combination, multiple simultaneous invocations of the *Group Management FU* for any given group are not allowed.

A *Group Management FU* shall not be invoked while a *Group Definition FU* is running for the group involved.

A *Group Management FU* shall not be invoked while a *Group Management FU* invocation or a *Group Definition FU* invocation is running for another group on another association with the same partner⁴⁵⁾.

⁴⁴⁾ Subject to local decision in the INITIATOR UE.

⁴⁵⁾ The purpose of this rule is to ensure the integrity of the configuration integrity control parameter for the INITIATOR system at the RESPONDER system. This parameter is updated by both the *Group Management FU* and the *Group Definition FU*; see 5.4.

6.2.2.3.3 Invoking events

Il est autorisé d'invoquer la partie INITIATOR de la *Group Management FU* par:

- une demande locale par le biais de la Coordinating Function, dont l'origine est en dehors du domaine d'application de ce rapport;
- la *Group Configuration FU*.

L'invocation de la partie RESPONDER de la *Group Management FU* est tentée chaque fois qu'une primitive *GroupMgmtInd* valide est reçue par le biais d'une association ayant les caractéristiques définies pour la *Group Management FU*.

6.2.2.4 Fin

6.2.2.4.1 Orderly termination

Il est autorisé de déclencher la fin ordonnée de la partie INITIATOR de l'invocation d'une *Group Management FU* par:

- une erreur d'encombrement. Voir 6.2.3.2.5;
- la réception d'une primitive de service *GroupMgmtConf* après l'émission d'une primitive de service *GroupMgmtReq* (fin normale).

La partie RESPONDER de l'invocation d'une *Group Management FU* se termine toujours d'elle-même de manière ordonnée (fin normale) après avoir émis une primitive de service *GroupMgmtResp*.

6.2.2.4.2 Disruption

Il est autorisé d'interrompre les parties INITIATOR et RESPONDER de l'invocation d'une *Group Management FU* par:

- l'interruption d'une autre invocation de FU. Voir 6.2.2.2.3;
- une condition d'erreur fatale rencontrée en fonctionnement. Voir 6.2.3.2.

6.2.3 Procédures

6.2.3.1 Primitives de service TASE.1

La primitive de service TASE.1 élémentaire suivante est utilisée par la *Group Management FU*: *GroupMgmt*.

6.2.3.1.1 Sequence

La séquence normale des primitives est la suivante:

INITIATOR UE	TASE.1	RESPONDER UE
<i>GroupMgmtReq</i>	----->	<i>GroupMgmtInd</i>
<i>GroupMgmtConf</i>	<-----	<i>GroupMgmtResp</i>

Règles:

- a) A réception d'une primitive de service *GroupMgmtInd* valide, le RESPONDER UE doit:
 - mettre à jour les attributs de groupe du CS(R) avec les informations de la primitive:

6.2.2.3.3 Invoking events

The INITIATOR part of the *Group Management FU* may be invoked by:

- local request via the Coordinating Function, the original source of which is outside the scope of this report;
- the *Group Configuration FU*.

Invocation of the RESPONDER part of the *Group Management FU* is attempted whenever a valid *GroupMgmtInd* primitive is received via an association with the characteristics as defined for the *Group Management FU*.

6.2.2.4 Termination

6.2.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of a *Group Management FU* invocation may be triggered by:

- congestion error. See 6.2.3.2.5;
- reception of a *GroupMgmtConf* service primitive after issuing a *GroupMgmtReq* service primitive (normal termination).

The RESPONDER part of a *Group Management FU* invocation always terminates itself in an orderly manner (normal termination) after issuing a *GroupMgmtResp* service primitive.

6.2.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Group Management FU* invocation may be disrupted by:

- disruption of another FU invocation. See 6.2.2.2.3;
- a fatal error condition encountered during operation. See 6.2.3.2.

6.2.3 Procedures

6.2.3.1 TASE.1 service primitives

The following elementary TASE.1 service is used by the *Group Management FU*:

GroupMgmt

6.2.3.1.1 Sequence

The normal sequence of primitives is as follows:

INITIATOR UE	TASE.1	RESPONDER UE
<i>GroupMgmtReq</i>	----->	<i>GroupMgmtInd</i>
<i>GroupMgmtConf</i>	<-----	<i>GroupMgmtResp</i>

Rules:

- a) Upon reception of a valid *GroupMgmtInd* service primitive, the RESPONDER UE shall:
 - Update group attributes in the CS(R) with information from the primitive:

Attribute	Parameter	Function
Group type	Group-type	<i>G-Create</i>
Group number	Group-number (Gnr)	<i>G-Create, G-Delete, Delete-all-groups</i>
Persistent	Persist	<i>G-Create, G-Change⁴⁶⁾</i>
Static	Static	<i>G-Create, G-Change</i>
Priority	Priority	<i>G-Create, G-Change</i>
Group size	MaxObj	<i>G-Create</i>
Object identifier size	MaxObjSize	<i>G-Create</i>

- générer une nouvelle valeur du paramètre de contrôle de l'intégrité de la configuration et le stocker parmi les données propres à l'INITIATOR UE actuel dans le système RESPONDER⁴⁷⁾;
 - renvoyer à l'INITIATOR UE la nouvelle valeur du paramètre de contrôle de l'intégrité de la configuration, par le biais du paramètre CE de la primitive de service *GroupMngtResp*.
- b) Le RESPONDER UE doit renvoyer l'état «ok» (paramètre *Result=result-ok*) si et seulement si la valeur mise à jour du paramètre de contrôle de l'intégrité de la configuration est ok.
- c) A réception d'une primitive de service *GroupMngtConf* valide, l'INITIATOR UE doit stocker la valeur du paramètre de contrôle de l'intégrité de la configuration reçue de la primitive parmi les données propres au RESPONDER UE en fonctionnement, dans le système INITIATOR⁴⁸⁾.
- d) Les règles énoncées en 6.1 s'appliquent également ici.
- e) Les règles énoncées dans la procédure de mise à jour du Configuration Set, en 5.4 , s'appliquent également.

⁴⁶⁾ Le passage de l'attribut "Persistent" de *true* à *false* n'est pas autorisé. Seul le passage de *false* à *true* est autorisé.

⁴⁷⁾ Voir la description de l'APDU pour ce qui concerne le paramètre de contrôle de l'intégrité de la configuration et les fonctions associées.

⁴⁸⁾ L'INITIATOR UE vérifiera par la suite sa valeur du paramètre de contrôle de l'intégrité de la configuration par rapport à la valeur véhiculée par la primitive *TASEBindConf*, lors des invocations ultérieures de la *Permanent Association FU* ou de la *Dynamic Association FU* par rapport au même système RESPONDER. Cela a pour but de déterminer si la configuration du groupe est à considérer comme encore valide ou non: voir article 5.

Attribute	Parameter	Function
Group type	Group-type	<i>G-Create</i>
Group number	Group-number (Gnr)	<i>G-Create, G-Delete, Delete-all-groups</i>
Persistent	Persist	<i>G-Create, G-Change</i> ⁴⁶⁾
Static	Static	<i>G-Create, G-Change</i>
Priority	Priority	<i>G-Create, G-Change</i>
Group size	MaxObj	<i>G-Create</i>
Object identifier size	MaxObjSize	<i>G-Create</i>

- Generate new value of the configuration integrity control parameter and store it as part of the data specific for the current INITIATOR UE, in the RESPONDER system⁴⁷⁾.
 - Report the new value of the configuration integrity control parameter back to the INITIATOR UE, via the CF parameter in the *GroupMgmtResp* service primitive.
- b) The RESPONDER UE shall return "OK" status (parameter Result = *result-ok*) if and only if the configuration integrity control parameter has been updated OK.
- c) Upon reception of a valid *GroupMgmtConf* service primitive, the INITIATOR UE shall store the value of the configuration integrity control parameter received from the primitive as part of the data specific for the current RESPONDER UE, in the INITIATOR system⁴⁸⁾.
- d) The rules stated in 6.1, above, also apply here.
- e) The rules stated in the procedure for Configuration Set update, in 5.4, also apply.

⁴⁶⁾ The attribute "Persistent" is not allowed to change from *true* to *false*, only from *false* to *true*.

⁴⁷⁾ See the APDU description, about the configuration integrity control parameter and associated functions.

⁴⁸⁾ The INITIATOR UE will later check his value of the configuration integrity control parameter against the value conveyed by the *TASEBindConf* primitive, during subsequent *Permanent Association FU* or *Dynamic Association FU* invocations against the same RESPONDER system. This is done in order to determine whether the group configuration is still to be considered valid or not: see clause 5.

6.2.3.1.2 Parameter values

GroupMgmt:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Fonction	<i>G-Create, G-Change, G-Delete</i> ou <i>Delete-all-groups</i> , selon la fonction par laquelle la FU est invoquée	Copie de la valeur à partir de <i>Ind</i>
Gtype (Group type)	<i>Valeur Measurand-group, Status-group, Discrete-group, Logical-breaker-status-group, Binary-command-group, Analog-setpoint-group, Digital-setpoint-group</i> ou <i>Text-message-group</i> . Fonction = <i>G-Create</i> : Valeur de l'attribut Group type à inclure dans le CS(R) pour le n° de groupe Group-number. Fonction = <i>G-Change</i> : La valeur doit être égale à la valeur de l'attribut Group type du CS(R) pour le n° de groupe Group number, pour que la demande soit légale. Fonction = <i>G-Delete</i> ou <i>Delete-all-groups</i> : La valeur est non pertinente.	Copie de la valeur à partir de <i>Ind</i>
Gnr (Group-number)	Numéro de référence pour le groupe en question. Fonction = <i>G-Create</i> : Valeur de l'attribut Group number dans le jeu d'attributs à créer dans le CS(R). Fonction = <i>G-Change</i> : Valeur de l'attribut Group number dans le jeu d'attributs du CS(R), pour lequel les attributs sont à modifier Fonction = <i>G-Delete</i> : Valeur de l'attribut Group number dans le jeu d'attributs à retirer du CS(R). Fonction = <i>Delete-all-groups</i> : La valeur est non pertinente.	Copie de la valeur à partir de <i>Ind</i>
CF	(Non applicable)	Valeur du Control Field pour l'INITIATOR UE qui a invoqué la Group Management FU. Voir 5.4.
MaxObj	Nombre maximal d'objets du groupe. Fonction = <i>G-Create</i> : Valeur de l'attribut Group size à inclure dans le CS(R) pour le n° de groupe Group-number. Fonction = <i>G-Change</i> : Pour que la demande soit légale, la valeur doit être égale à la valeur de l'attribut Group size du CS(R) pour le n° de groupe Group-number. Fonction = <i>G-Delete</i> ou <i>Delete-all-groups</i> : La valeur est non pertinente.	(Non applicable)
MaxObjSize	Taille de l'identificateur Max Object (en octets). Fonction = <i>G-Create</i> : Valeur de l'attribut Object identifier size à inclure dans le CS(R) pour le n° de groupe Group-number. Fonction = <i>G-Change</i> : Pour que la demande soit légale, la valeur doit être égale à la valeur de l'attribut Object identifier size du CS(R) pour le n° de groupe Group-number. Fonction = <i>G-Delete</i> ou <i>Delete-all-groups</i> : La valeur est non pertinente.	(Non applicable)

6.2.3.1.2 Parameter values

GroupMgmt:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Function	<i>G-Create</i> , <i>G-Change</i> , <i>G-Delete</i> or <i>Delete-all-groups</i> , according to the function with which the FU is invoked	Copy of value from <i>Ind</i>
Gtype (Group type)	Value <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> , <i>Binary-command-group</i> , <i>Analog-setpoint-group</i> , <i>Digital-setpoint-group</i> or <i>Text-message-group</i> . Function = <i>G-Create</i> : Value of attribute Group type to be included in the CS(R) for group No. Group-number. Function = <i>G-Change</i> : Value shall be equal to value of attribute Group type in the CS(R) for group No. Group-number, for the request to be legal. Function = <i>G-Delete</i> or <i>Delete-all-groups</i> : Value is irrelevant.	Copy of value from <i>Ind</i>
Gnr (Group-number)	Reference number for the group in question. Function = <i>G-Create</i> : Value of attribute Group number in set of attributes to be created in the CS(R). Function = <i>G-Change</i> : Value of attribute Group number in set of attributes in the CS(R), for which attributes are to be modified Function = <i>G-Delete</i> : Value of attribute Group number in set of attributes to be removed from the CS(R). Function = <i>Delete-all-groups</i> : Value is irrelevant.	Copy of value from <i>Ind</i>
CF	(Not applicable)	Control Field value for the INITIATOR UE that invoked the <i>Group Management FU</i> . See 5.4.
MaxObj	Maximum number of objects in group. Function <i>G-Create</i> : Value of attribute Group size to be included in the CS(R) for group No. Group-number. Function = <i>G-Change</i> : Value shall be equal to value of attribute Group size in the CS(R) for group No. Group-number, for the request to be legal. Function = <i>G-Delete</i> or <i>Delete-all-groups</i> : Value is irrelevant.	(Not applicable)
MaxObjSize	Max Object identifier size (in octets). Function <i>G-Create</i> : Value of attribute Object identifier size to be included in the CS(R) for group No. Group-number. Function = <i>G-Change</i> : Value shall be equal to value of attribute Object identifier size in the CS(R) for group No. Group-number, for the request to be legal. Function = <i>G-Delete</i> or <i>Delete-all-groups</i> : Value is irrelevant.	(Not applicable)

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Persist	Drapeau persistant. Function = <i>G-Create</i> ou <i>G-Change</i> : Valeur de l'attribut Persistent à inclure dans le CS(R) pour le n° de groupe Group-number. Function = <i>G-Delete</i> ou <i>Delete-all-groups</i> : La valeur est non pertinente.	(Non applicable)
Static	Drapeau statique. Function = <i>G-Create</i> ou <i>G-Change</i> : Valeur de l'attribut Static à inclure dans le CS(R) pour le n° de groupe Group-number. Function = <i>G-Delete</i> ou <i>Delete-all-groups</i> : La valeur est non pertinente.	(Non applicable)
Priority	Valeur de priorité. Function = <i>G-Create</i> ou <i>G-Change</i> : Valeur de l'attribut Priority class à inclure dans le CS(R) pour le n° de groupe Group-number. Function = <i>G-Delete</i> ou <i>Delete-all-groups</i> : La valeur est non pertinente.	(Non applicable)
Result	(Non applicable)	Action effectuée comme spécifié. = <i>result-ok</i> Action non effectuée du fait d'une condition d'erreur: Autre valeur. Voir 6.2.3.2.

Noter qu'avec Function = *G-Change*, le RESPONDER UE doit être en mesure de traiter la modification d'un ou plusieurs des attributs Persistent, Static et Priority class pendant une seule transaction *GroupMgmt*.

6.2.3.2 Traitement des erreurs

6.2.3.2.1 FU disruption

Interruption par la *Permanent Association FU* ou la *Dynamic Association FU*:

L'interruption de la partie INITIATOR et de la partie RESPONDER de l'invocation en cours de la *Group Management FU* doit être déclenchée localement, dans le cadre du traitement des primitives *Abort Ind* entrantes de la partie INITIATOR et de la partie RESPONDER de l'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* qui traite l'association où l'invocation de la *Group Management FU* est en fonctionnement.

Les deux parties de l'invocation en cours de la *Group Management FU* doivent être terminées correctement, sans que l'une ou l'autre des parties n'essaie d'émettre une primitive associée à la fin elle-même⁴⁹⁾.

⁴⁹⁾ Les modes opératoires locaux de nettoyage ne sont pas spécifiés dans le présent rapport technique.

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Persist	Persistent flag. Function = <i>G-Create</i> or <i>G-Change</i> : Value of attribute Persistent to be included in the CS(R) for group No. Group-number. Function = <i>G-Delete</i> or <i>Delete-all-groups</i> : Value is irrelevant.	(Not applicable)
Static	Static flag. Function = <i>G-Create</i> or <i>G-Change</i> : Value of attribute Static to be included in the CS(R) for group No. Group-number. Function = <i>G-Delete</i> or <i>Delete-all-groups</i> : Value is irrelevant.	(Not applicable)
Priority	Priority value. Function = <i>G-Create</i> or <i>G-Change</i> : Value of attribute Priority class to be included in the CS(R) for group No. Group-number. Function = <i>G-Delete</i> or <i>Delete-all-groups</i> : Value is irrelevant.	(Not applicable)
Result	(Not applicable)	Action performed as specified. = <i>result-ok</i> Action not performed, due to error condition: Other value. See 6.2.3.2.

Note that with Function = *G-Change*, the RESPONDER UE shall be able to handle modification of one or more of the attributes Persistent, Static and Priority class during a single *GroupMgmt* transaction.

6.2.3.2 Error handling

6.2.3.2.1 FU disruption

Disruption by the *Permanent Association FU* or *Dynamic Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Group Management FU* shall be triggered locally, as a part of the handling of incoming *AbortInd* primitives in both the INITIATOR part and the RESPONDER part of the *Permanent Association FU* or *Dynamic Association FU* invocation handling the association where the *Group Management FU* invocation is running.

Both parts of the current invocation of the *Group Management FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself⁴⁹⁾

⁴⁹⁾ Local clean-up procedures are not specified by this technical report.

6.2.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Group Management FU* n'est pas présente dans un INITIATOR UE:

Le traitement de ce type d'erreur résultant d'une tentative d'invocation par la Coordinating Function locale constitue un problème local qui ne fait pas partie du domaine d'application du présent rapport technique.

Si ce type d'erreur survient lors d'une tentative d'invocation du fait d'une invocation de la *Group Configuration FU*, l'erreur est traitée par le mécanisme de traitement des erreurs de la *Group Configuration FU*. Voir 6.5.

Si la *Group Management FU* n'est pas présente dans un RESPONDER UE:

Le RESPONDER UE doit répondre aux tentatives d'activation de la manière suivante:

- émettre une primitive *GroupMgmtResp* avec Result = *remote-service-user-unavailable*.

FU présente, mais tentative illégale:

Les tentatives d'invocation qui violent les règles indiquées en 6.2.2.3.2 dans un INITIATOR UE constituent un problème local qui ne fait pas partie du domaine d'application du présent rapport technique.

Dans un RESPONDER UE, les tentatives d'invocation qui violent les mêmes règles doivent être traitées par le RESPONDER UE de la manière suivante:

- émission d'une primitive *GroupMgmtResp* avec Result = *remote-service-user-unavailable*, sans (ré)invocation de la *Group Management FU* pour le groupe concerné, dans le RESPONDER UE.

6.2.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>GroupMgmtConf</i>
FU non en fonctionnement	Ignorer, ou Indication d'erreur locale
FU en fonctionnement, en attente de <i>GroupMgmtConf</i>	(Normal)

Partie RESPONDER:

Non applicable. Une primitive de service *GroupMgmtInd* n'est jamais hors contexte dans le RESPONDER⁵⁰⁾.

6.2.3.2.4 Timing errors

Erreur dans la partie INITIATOR:

Non applicable. Aucune contrainte de synchronisation n'est imposée à la partie INITIATOR de la *Group Management FU*.

⁵⁰⁾ L'ordonnancement correct de la primitive *Ind/Resp* est imposé par le TASE.

6.2.3.2.2 Illegal invocation attempt

FU not present:

If the *Group Management FU* is not present in an INITIATOR UE:

The handling of this type of error resulting from an invocation attempt by the local Coordinating Function is a local issue, outside the scope of this technical report.

If this type of error occurs upon an invocation attempt by a *Group Configuration FU* invocation, the error is handled by the error handling mechanism within the *Group Configuration FU*. See 6.5.

If the *Group Management FU* is not present in a RESPONDER UE:

The RESPONDER UE shall respond to activation attempts in the following way:

- issuing a *GroupMgmtResp* primitive with Result = *remote-service-user-unavailable*

FU present, but attempt illegal:

Invocation attempts violating the rules stated in 6.2.2.3.2 in an INITIATOR UE is a local issue, outside the scope of this technical report.

In a RESPONDER UE, invocation attempts violating the same rules shall be handled by the RESPONDER UE in the following way:

- issuing a *GroupMgmtResp* primitive with Result = *remote-service-user-unavailable*, without (re-)invoking the *Group Management FU* for the group concerned, in the RESPONDER UE.

6.2.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>GroupMgmtCnf</i>
FU not running	Ignore, or Local error indication
FU running, waiting for <i>GroupMgmtConf</i>	(Normal)

RESPONDER part:

Not applicable. A *GroupMgmtInd* service primitive is never out of context in the RESPONDER⁵⁰⁾.

6.2.3.2.4 Timing errors

Error in INITIATOR part:

Not applicable. There are no timing restraints placed upon the INITIATOR part of the *Group Management FU*.

⁵⁰⁾ Proper *Ind/Resp* primitive sequencing is enforced by the TASE.

Erreur dans la partie RESPONDER:

Erreur	Réaction de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à <i>GroupMgmtInd</i>	Sur le RESPONDER: Erreur locale lors d'une éventuelle tentative d'émission de <i>GroupMgmtResp</i>	Partie RESPONDER: Terminer l'invocation de la FU localement .
	Sur l'INITIATOR: <i>GroupMgmtConf</i> , avec Result = <i>remote-service-user-unavailable</i> .	Partie INITIATOR: Terminer l'invocation de la FU localement .

6.2.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsqu'elle se produit lors d'une tentative de *GroupMgmtReq*:

- terminer l'invocation de la FU localement .

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsqu'elle se produit lors d'une tentative de *GroupMgmtResp*:

- terminer l'invocation de la FU localement . (La partie INITIATOR de la FU sera finalement terminée après expiration du délai du fournisseur de TASE.1.)

Voir 4.2.1.

6.2.3.2.6 TASE.1 service primitive parameters errors

Erreurs dans la primitive *GroupMgmtInd* (détectées par la partie RESPONDER):

- La partie RESPONDER doit dans tous les cas de ce type émettre une *GroupMgmtResp* avec Result = <Valeur du tableau ci-dessous>, et les valeurs de Group-number et de Group-type comme dans le *Ind* correspondant.
- La FU ne doit pas être invoquée.

Erreur	Valeur du paramètre Result
Function = <i>G-Create</i> et valeur de Group-type autre que: <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> , <i>Binary-command-group</i> , <i>Analog-setpoint-group</i> , <i>Digital-setpoint-group</i> ou <i>Text-message-group</i>	<i>gtype-out-of-range</i>
Function = <i>G-Create</i> , et: - Group-number < 1, ou - Group-number est supérieur à la plus grande valeur de l'attribut de groupe Group number acceptable dans le système RESPONDER	<i>gnr-out-of-range</i>
Function = <i>G-Create</i> , et le n° de groupe Group-number existe déjà dans CS(R).	<i>group-exists</i>
Function = <i>G-Create</i> , et un nouveau descripteur de groupe ne peut pas être alloué à cause des limitations des ressources locales	<i>no-memory</i>
Function = <i>G-Create</i> et: - Gsize < 1, ou - Gsize est supérieur à la plus grande valeur de l'attribut de groupe Group size acceptable dans le système RESPONDER	<i>gsize-out-of-range</i>

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>GroupMgmtInd</i>	In RESPONDER: Local error from eventual attempt in issuing <i>GroupMgmtResp</i> In INITIATOR: <i>GroupMgmtConf</i> , with Result = <i>remote-service-user-unavailable</i> .	RESPONDER part: Terminate FU invocation locally. INITIATOR part: Terminate FU invocation locally.

6.2.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with a *GroupMgmtReq* attempt:

- Terminate FU invocation locally.

See 4.2.1.

RESPONDER part of the FU:

When occurring with a *GroupMgmtResp* attempt:

- Terminate FU invocation locally. (INITIATOR part of FU will eventually be terminated after timeout in the TASE.1 provider.)

See 4.2.1.

6.2.3.2.6 TASE.1 service primitive parameter errors

Errors in the *GroupMgmtInd* primitive (detected by the RESPONDER part):

The RESPONDER part shall in all these cases issue a *GroupMgmtResp* with Result = <Value from table below>, and the values of Group-number and Group-type as in the corresponding *Ind*

The FU shall not be invoked.

Error	Value of parameter Result
Function = <i>G-Create</i> , and Group-type value other than: <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> , <i>Binary-command-group</i> , <i>Analog-setpoint-group</i> , <i>Digital-setpoint-group</i> or <i>Text-message-group</i>	<i>gtype-out-of-range</i>
Function = <i>G-Create</i> , and: – Group-number < 1, or – Group-number greater than greatest value of the group attribute Group number that is acceptable in the RESPONDER system	<i>gnr-out-of-range</i>
Function = <i>G-Create</i> , and group No. Group-number already exists in CS(R)	<i>group-exists</i>
Function = <i>G-Create</i> , and a new group descriptor cannot be allocated because of local resource limitations	<i>no-memory</i>
Function = <i>G-Create</i> , and: – Gsize < 1, or – Gsize greater than greatest value of the group attribute Group size that is acceptable in the RESPONDER system	<i>gsize-out-of-range</i>

Erreur	Valeur du paramètre Result
Function = <i>G-Create</i> , et: - Objlength < 1, ou - Objlength est supérieur à la plus grande valeur de l'attribut Object identifier size acceptable dans le système RESPONDER	<i>objlength-out-of-range</i>
Function = <i>G-Create</i> ou <i>G-Change</i> , et: - Priority < 0 ou > 15, ou - Priority différent de 0, et le mécanisme de priorité n'est pas supporté par le RESPONDER UE.	<i>priority-out-of-range</i>
Function = <i>G-Change</i> et la valeur de Group-type n'est pas égale à la valeur de l'attribut Group type du CS(R) pour le n° de groupe Group-number. (Le cas du numéro de groupe Group-number non existant est traité ci-dessous.)	<i>gtype-out-of-range</i>
Function = <i>G-Change</i> et valeur de Gsize n'est pas égale à la valeur de l'attribut Group size du CS(R) pour le n° de groupe Group-number ⁵¹⁾ . (Le cas du numéro de groupe Group-number non existant est traité ci-dessous.)	<i>gsize-out-of-range</i>
Function = <i>G-Change</i> , et la valeur d'Objlength n'est pas égale à la valeur de l'attribut Object identifier size du CS(R) pour le n° de groupe Group-number ⁵²⁾ . (Le cas du numéro de groupe Group-number non existant est traité ci-dessous.)	<i>objlength-out-of-range</i>
Function = <i>G-Change</i> , et le paramètre Persist = <i>false</i> , et l'attribut Persistent = <i>true</i> pour le groupe dans le CS(R) ⁵³⁾	<i>not-deletable</i>
Function = <i>G-Change</i> ou <i>G-Delete</i> ⁵⁴⁾ , et le n° de groupe Group-number n'existe pas dans le CS(R).	<i>gnr-out-of-range</i>
Function = <i>G-Delete</i> , et l'attribut Persistent = <i>true</i> pour le groupe dans le CS(R)	<i>not-deletable</i>
Le RESPONDER UE n'est pas en mesure d'exécuter la fonction demandée pour une raison autre que celles énumérées ci-dessus	<i>remote-service-user-unavailable</i>

Noter que le RESPONDER UE n'est pas autorisé à rejeter une demande de suppression de TOUS les groupes non permanents du CS(R): une primitive *GroupMgmtInd* avec Function = *Delete-all-groups* doit toujours être valide. Elle a pour effet de supprimer tous les groupes éventuels du CS(R) pour lesquels Persistent = *false*.

⁵¹⁾ Equivaut à demander une modification de l'attribut Object Identifier size pour un groupe du CS(R). Cela est illégal: voir 9.4 de la CEI 60870-6-501.

⁵²⁾ Equivaut à demander une modification de l'attribut Group size pour un groupe du CS(R). Cela est illégal: voir 9.4 de la CEI 60870-6-501.

⁵³⁾ Equivaut à demander une modification de la valeur de l'attribut Persistent de *true* à *false*, ce qui doit être illégal.

⁵⁴⁾ Un groupe non existant à supprimer est considéré comme une erreur à ce stade. Toutefois, s'il est souhaitable de traiter comme correct le résultat erroné d'une tentative de suppression d'un groupe non existant, l'INITIATOR UE est parfaitement autorisé à le faire dans le cadre du traitement de cette erreur de la *Group Management FU* à un niveau supérieur, qui ne fait pas partie du domaine d'application de ce rapport.

Error	Value of parameter Result
Function = <i>G-Create</i> , and: – Objlength < 1, or – Objlength greater than greatest value of the attribute Object identifier size that is acceptable in the RESPONDER system	<i>objlength-out-of-range</i>
Function = <i>G-Create</i> or <i>G-Change</i> , and: – Priority < 0 or > 15, or – Priority not equal to 0, and priority mechanism not supported by the RESPONDER UE.	<i>priority-out-of-range</i>
Function = <i>G-Change</i> , and Group-type value not equal to value of attribute Group type in CS(R) for group No. Group-number. (The case of non-existing group No. Group-number is considered below.)	<i>gtype-out-of-range</i>
Function = <i>G-Change</i> , and Gsize value not equal to value of attribute Group size in CS(R) for group No. Group-number. ⁵¹⁾ (The case of non-existing group No. Group-number is considered below.)	<i>gsize-out-of-range</i>
Function = <i>G-Change</i> , and Objlength value not equal to value of attribute Object identifier size in CS(R) for group No. Group-number. ⁵²⁾ (The case of non-existing group No. Group-number is considered below.)	<i>objlength-out-of-range</i>
Function = <i>G-Change</i> , and parameter Persist = <i>false</i> , and attribute Persistent = <i>true</i> for the group in CS(R) ⁵³⁾	<i>not-deletable</i>
Function = <i>G-Change</i> or <i>G-Delete</i> ⁵⁴⁾ , and group No. Group-number does not exist in the CS(R).	<i>gnr-out-of-range</i>
Function = <i>G-Delete</i> , and attribute Persistent = <i>true</i> for the group in CS(R)	<i>not-deletable</i>
RESPONDER UE not able to perform the requested <i>unavailable</i> function, due to any reason other than those listed above	<i>remote-service-user-unavailable</i>

Note that the RESPONDER UE is not allowed to reject a request for deleting ALL non-permanent groups in the CS(R): A *GroupMgmtInd* primitive with Function = *Delete-all-groups* shall always be valid, having the effect of deleting all groups in the CS(R), if any, for which Persistent = *false*.

⁵¹⁾ Equivalent to requesting a change of attribute Object identifier size for a group within the CS(R). This is illegal; see 9.4 of IEC 60870-6-501.

⁵²⁾ Equivalent to requesting a change of attribute Group size for a group within the CS(R). This is illegal; see 9.4 of IEC 60870-6-501.

⁵³⁾ Equivalent to requesting a change in value of attribute Persistent from *true* to *false*, which shall be illegal.

⁵⁴⁾ Non-existent group to be deleted is considered an error at this stage. However, if it is desirable to treat the erroneous result of attempting to delete a non-existing group as "OK", the INITIATOR UE may very well do so, in the course of handling this *Group Management FU* error at a higher level, not within the scope of this report.

Erreurs dans la primitive *GroupMgmtConf* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Absence de correspondance entre Group-type dans la primitive et Group-type dans la <i>GroupMgmtReq</i> correspondante, et Result = <i>result-ok</i> .	Terminer l'invocation de la FU localement , en enregistrant l'erreur. Mettre à jour le paramètre de contrôle d'intégrité de la configuration de manière normale ⁵⁵⁾
Result >< <i>result-ok</i>	Terminer l'invocation de la FU localement , en enregistrant l'erreur. Ne pas mettre à jour le paramètre de contrôle d'intégrité de la configuration.

6.3 Group Definition FU

Type: Primary.

6.3.1 Fonction

L'invocation de la *Group Definition FU* autorise l'INITIATOR UE qui exécute l'invocation à modifier l'attribut Object set (paires d'attributs Object number et Object identifier) du descripteur de groupe qui contient une valeur donnée de l'attribut Group number, dans le CS(R) d'un RESPONDER UE.

Cela équivaut à définir ou redéfinir les objets d'un groupe du CS(R).

Il est autorisé d'invoquer la FU soit pour une définition, soit pour une redéfinition:

Dans le cas d'une définition, une mise à jour de l'Object set tout entier du groupe doit être tentée.

Dans le cas d'une redéfinition, une mise à jour d'un nombre spécifié d'Object identifiers du groupe doit être tentée. La partie à mettre à jour est spécifiée par une plage d'Object number ayant une étendue maximale définie; voir ci-dessous les règles d'utilisation des paramètres de primitive de service Index1 et Index2. Les Object numbers eux-mêmes ne sont pas susceptibles d'être modifiés pendant une redéfinition.

Les parties INITIATOR et RESPONDER de cette FU sont généralement asymétriques par rapport aux événements d'invocation et de fin, lorsqu'elles sont invoquées pour une définition:

La partie RESPONDER n'a aucun moyen de savoir si une transaction *DefineGroup* terminée est la dernière d'une série de transactions constituant l'invocation de la *Group Definition FU*; voir 6.3.3.1.1. En conséquence, la partie RESPONDER est généralement invoquée un certain nombre de fois, une fois pour chaque transaction *DefineGroup* entrante. La partie INITIATOR, au contraire, n'est invoquée qu'une seule fois par invocation de la *Group Definition FU* proprement dite. Il convient donc de comprendre «invocation de la *Group Definition FU*» comme signifiant strictement «partie INITIATOR de l'invocation de la *Group Definition FU*». C'est toutefois la formulation «invocation de la *Group Definition FU*» qui est utilisée tout au long du présent rapport technique.

⁵⁵⁾ Etant donné que Result = *result-ok*, le paramètre de contrôle de l'intégrité de la configuration aura été mis à jour dans le système RESPONDER.

Errors in the *GroupMgmtConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Mismatch between Group-type in primitive and Group-type in corresponding <i>GroupMgmtReq</i> , and Result = <i>result-ok</i> .	Terminate the FU invocation locally, registering the error. Update the configuration integrity control parameter, as normal ⁵⁵⁾
Result >< <i>result-ok</i>	Terminate the FU invocation locally, registering the error. Do not update the configuration integrity control parameter.

6.3 Group Definition FU

Type: Primary.

6.3.1 Function

A *Group Definition FU* invocation allows the invoking INITIATOR UE to modify the attribute Object set (component attribute pairs Object number and Object identifier) within the group descriptor that contains a given value of attribute Group number, in the CS(R) of a RESPONDER UE.

This is equivalent to defining or redefining objects in a group in the CS(R).

The FU may be invoked for either a definition or a redefinition:

In the case of a definition, an update of the whole Object set of the group shall be attempted.

In the case of a redefinition, an update of a specified number of Object identifiers within the group shall be attempted. The part that is to be updated is specified by an Object number range with a defined maximum span; see below for rules on using the service primitive parameters Index1 and Index2. Object numbers themselves cannot be modified during a redefinition.

The INITIATOR and RESPONDER parts of this FU are generally asymmetrical with regard to invocation and termination events, when invoked for definition:

The RESPONDER part has no way of knowing whether a completed *Define Group* transaction is the last in the series of transactions constituting the *Group Definition FU* invocation; see 6.3.3.1.1. Consequently, the RESPONDER part is generally invoked a number of times, once for each incoming *Define Group* transaction. The INITIATOR part, on the other hand, is invoked only once per invocation of the *Group Definition FU* itself. The term "*Group Definition FU* invocation" should therefore strictly read "INITIATOR part of *Group Definition FU* invocation". However, "*Group Definition FU* invocation" is used throughout this technical report.

⁵⁵⁾ Because Result = *result-ok*, the configuration integrity control parameter will have been updated in the RESPONDER system.

6.3.2 Règles de coordination

6.3.2.1 Utilisation d'une association

Toutes les interactions TASE.1 faisant partie d'une invocation de la *Group Definition FU* sont véhiculées par une seule association. L'association doit posséder les caractéristiques spécifiées en 6.3.2.3.1.

6.3.2.2 Relations avec les autres FU

6.3.2.2.1 Invoking FUs

La *Group Configuration FU* est autorisée à invoquer la *Group Definition FU*.

6.3.2.2.2 Invoked FUs

La *Group Definition FU* ne doit invoquer aucune autre FU.

6.3.2.2.3 Disrupting FUs

Il est admis d'interrompre la *Group Definition FU* par:

- la *Permanent Association FU*;
- la *Dynamic Association FU*.

6.3.2.2.4 Disrupted FUs

Il est admis que la *Group Definition FU* interrompe les FU suivantes:

- *Group Configuration FU*;
- *Requested Data Transfer FU*;
- *Periodically Requested Data Transfer FU*;
- *Unsolicited Data Transfer FU*;
- *Unsolicited Periodic Data Transfer FU*;
- *Unsolicited Spontaneous Data Transfer FU*;
- *Supervisory Control Data Transfer FU*.

6.3.2.3 Invocation

6.3.2.3.1 Conditions préalables

L'une⁵⁶⁾ des FU suivantes doit avoir été invoquée avant toute invocation de la *Group Definition FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*.

L'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* doit encore être en fonctionnement au moment de l'invocation d'une *Group Definition FU*. Dans le cas de la *Permanent Association FU*, l'association maintenue par l'invocation doit être en fonctionnement (et non interrompue temporairement) à ce moment-là.

⁵⁶⁾ Sous réserve d'une décision locale dans l'INITIATOR UE.

6.3.2 Coordination rules

6.3.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Group Definition FU* are conveyed by one single association. The association shall have the characteristics as specified in 6.3.2.3.1.

6.3.2.2 Relation to other FUs

6.3.2.2.1 Invoking FUs

The *Group Definition FU* may be invoked by the *Group Configuration FU*.

6.3.2.2.2 Invoked FUs

The *Group Definition FU* shall not invoke any other FU.

6.3.2.2.3 Disrupting FUs

The *Group Definition FU* may be disrupted by:

- *Permanent Association FU*;
- *Dynamic Association FU*.

6.3.2.2.4 Disrupted FUs

The *Group Definition FU* may disrupt the following FUs:

- *Group Configuration FU*;
- *Requested Data Transfer FU*;
- *Periodically Requested Data Transfer FU*;
- *Unsolicited Data Transfer FU*;
- *Unsolicited Periodic Data Transfer FU*;
- *Unsolicited Spontaneous Data Transfer FU*;
- *Supervisory Control Data Transfer FU*.

6.3.2.3 Invocation

6.3.2.3.1 Prerequisites

One⁵⁶⁾ of the following FUs shall have been invoked preceding any invocation of the *Group Definition FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*.

The *Permanent Association FU* or *Dynamic Association FU* invocation shall still be running at the time a *Group Definition FU* is invoked. In the case of *Permanent Association FU* the association maintained by the invocation shall be running (not temporarily broken) at that time.

⁵⁶⁾ Subject to local decision in the INITIATOR UE.

La *Permanent Association FU* ou la *Dynamic Association FU* doivent avoir été invoquées afin de créer (et, pour la *Permanent Association FU*, également de maintenir) l'association à utiliser pour les interactions liées à l'invocation en cours de la *Group Definition FU*, avec la paire A-suffix spécifiée en 5.2.1.

6.3.2.3.2 Restrictions

Pour toute combinaison donnée de systèmes INITIATOR/RESPONDER, des invocations multiples simultanées de la *Group Definition FU* ne sont autorisées pour aucun groupe.

Une *Group Definition FU* ne doit pas être invoquée pendant qu'une invocation de la *Group Definition FU* ou de la *Group Management FU* est en fonctionnement pour un groupe quelconque, pour une association avec le même partenaire⁵⁷⁾.

6.3.2.3.3 Invoking events

Il est autorisé d'invoquer la partie INITIATOR de la *Group Definition FU* par:

- une demande locale par le biais de la Coordinating Function, dont l'origine est en dehors du domaine d'application du présent rapport technique;
- la *Group Configuration FU*.

L'invocation de la partie RESPONDER de la *Group Definition FU* est tentée chaque fois qu'une primitive *DefineGroupInd* valide est reçue par le biais d'une association avec les caractéristiques définies pour la *Group Definition FU*.

6.3.2.4 Fin

6.3.2.4.1 Orderly termination

Il est autorisé de déclencher la fin ordonnée de la partie INITIATOR de l'invocation d'une *Group Definition FU* par:

- une erreur d'encombrement. Voir 6.3.3.2.5;
- la réception d'une primitive de service *DefineGroupConf* non valide après l'émission d'une primitive de service *DefineGroupReq* (fin suite à une erreur);
- la réception d'une primitive de service *DefineGroupConf* valide après l'émission de la dernière primitive de service *DefineGroupReq* de l'invocation en cours de la *Group Definition FU* (fin normale).

La partie RESPONDER de l'invocation d'une *Group Definition FU* se termine toujours d'elle-même de manière ordonnée (fin normale) après avoir émis une primitive de service *DefineGroupResp*.

6.3.2.4.2 Disruption

Il est autorisé d'interrompre les parties INITIATOR et RESPONDER de l'invocation d'une *Group Definition FU* par:

- l'interruption d'une autre invocation de FU. Voir 6.3.2.2.3.

⁵⁷⁾ L'objet de cette règle est d'assurer l'intégrité du paramètre de contrôle de l'intégrité de la configuration pour le système INITIATOR au niveau du système RESPONDER. Ce paramètre est mis à jour par la *Group Management FU* et la *Group Definition FU*; voir 5.4.

The *Permanent Association FU* or *Dynamic Association FU* shall have been invoked in order to create (and, for the *Permanent Association FU*, also to maintain) the association to be used for the interactions related to the current invocation of the *Group Definition FU*, with the A-suffix pair specified in 5.2.1.

6.3.2.3.2 Restrictions

For any given INITIATOR/RESPONDER system combination, multiple simultaneous invocations of the *Group Definition FU* for any given group are not allowed.

A *Group Definition FU* shall not be invoked while a *Group Definition FU invocation* or a *Group Management FU invocation* is running for any group on an association with the same partner⁵⁷⁾.

6.3.2.3.3 Invoking events

The INITIATOR part of the *Group Definition FU* may be invoked by:

- Local request via the Coordinating Function, the original source of which is outside the scope of this technical report;
- the *Group Configuration FU*.

Invocation of the RESPONDER part of the *Group Definition FU* is attempted whenever a valid *DefineGroupInd* primitive is received via an association with the characteristics as defined for the *Group Definition FU*.

6.3.2.4 Termination

6.3.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of a *Group Definition FU* invocation may be triggered by:

- Congestion error. See 6.3.3.2.5.
- Reception of an invalid *DefineGroupConf* service primitive after issuing an *DefineGroupReq* service primitive (termination on error);
- Reception of a valid *DefineGroupConf* service primitive after issuing the last *DefineGroupReq* service primitive of the current *Group Definition FU* invocation (normal termination).

The RESPONDER part of a *Group Definition FU* invocation always terminates itself in an orderly manner (normal termination) after issuing a *DefineGroupResp* service primitive.

6.3.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Group Definition FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 6.3.2.2.3.

⁵⁷⁾ The purpose of this rule is to ensure the integrity of the configuration integrity control parameter for the INITIATOR system in the RESPONDER system. This parameter is updated by both the *Group Management FU* and the *Group Definition FU*; see 5.4.

6.3.3 Procédures

6.3.3.1 Primitives de service TASE.1

La primitive de service TASE.1 élémentaire suivante est utilisée par la *Group Definition FU*:

– *DefineGroup*

6.3.3.1.1 Séquence

A) Si l'invocation a pour but une redéfinition:

La séquence redefinition des primitives consiste en une seule transaction *DefineGroup*:

INITIATOR UE	TASE.1	RESPONDER UE
<i>DefineGroupReq</i>	----->	<i>DefineGroupInd</i>
<i>DefineGroupConf</i>	<-----	<i>DefineGroupResp</i>

Règles:

- La plage Object number ou subgroup couverte par la transaction est autorisée à comprendre de 1 à un maximum d'objets. Le maximum est fonction de la longueur des divers Object identifiants correspondants et il admis que ce maximum varie d'une transaction à l'autre.⁵⁸⁾
- Les Object identifiants pour TOUS les Object numbers du sous-groupe (voir la règle a)) couverts par une transaction doivent être mis à jour dans le CS(R) pendant la transaction.
- A réception d'une primitive de service *DefineGroupInd* valide, le RESPONDER UE doit:
 - Mettre à jour l'attribut de groupe Object set du CS(R) avec les informations de la primitive:
Les Object identifiants extraits à partir du paramètre Objid(I) pour les valeurs croissantes consécutives d'index (I)⁵⁹⁾ doivent être associés aux valeurs croissantes consécutives d'Object number dans le CS(R), en commençant par la valeur de Index1.
 - Générer une nouvelle valeur du paramètre de contrôle d'intégrité de la configuration et la stocker parmi les données propres à l'INITIATOR UE en fonctionnement, dans le système RESPONDER.⁶⁰⁾
 - Renvoyer la nouvelle valeur du paramètre de contrôle d'intégrité de la configuration à l'INITIATOR UE, par le biais du paramètre CF de la primitive de service *DefineGroupResp*.
- Le RESPONDER UE doit renvoyer l'état «ok» (paramètre Result(I)=*result-ok* pour tous les I) si et seulement si la valeur mise à jour du paramètre de contrôle d'intégrité de la configuration est ok.

⁵⁸⁾ Bien que le protocole ne limite pas la taille des données, il se peut que cela soit nécessaire à l'implémentation. Il convient que toute implémentation soit au moins en mesure de répondre avec des données conformes au document NIST SP 500-177 (voir bibliographie [1]).

⁵⁹⁾ Cet index n'est pas à interpréter comme un index d'un tableau Objid(I) au sens ordinaire, mais seulement comme un nombre entier d'ordonnement des éléments codés dans le paramètre Objid. Les éléments (Object identifiants) sont des chaînes de caractères ASCII, toutes codées consécutivement dans un tableau d'octets unidimensionnel ; voir A.2.4 de la CEI 60870-6-501. Il faut que l'augmentation de la valeur de "index I" corresponde au passage à une valeur d'index supérieure dans ce tableau d'octets.

⁶⁰⁾ Voir la description de l'APDU pour ce qui concerne le paramètre de contrôle de l'intégrité de la configuration et les fonctions associées.

6.3.3 Procedures

6.3.3.1 TASE.1 service primitives

The following elementary TASE.1 service primitive is used by the *Group Definition FU*:

– *DefineGroup*

6.3.3.1.1 Sequence

A) If invoked for redefinition:

The redefinition sequence of primitives is a single *DefineGroup* transaction:

INITIATOR UE	TASE.1	RESPONDER UE
<i>DefineGroupReq</i>	----->	<i>DefineGroupInd</i>
<i>DefineGroupConf</i>	<-----	<i>DefineGroupResp</i>

Rules:

- The Object number range, or subgroup, that the transaction covers, may span a number of objects from 1 up to a maximum. The maximum is a function of the individual lengths of the corresponding Object identifiers, and may vary between transactions⁵⁸⁾.
- Object identifiers for ALL Object numbers within the subgroup (see rule a)) that an individual transaction covers, shall be updated in the CS(R) during the transaction.
- Upon reception of a valid *DefineGroupInd* service primitive, the RESPONDER UE shall:
 - Update group attribute Object set in the CS(R) with information from the primitive:
The Object identifiers fetched from the parameter *Objid(I)* at consecutive ascending index (I)⁵⁹⁾ values shall be associated with consecutive ascending values of Object number in the CS(R), starting with the value of *Index1*.
 - Generate new value of the configuration integrity control parameter and store it as part of the data specific for the current INITIATOR UE, in the RESPONDER system⁶⁰⁾.
 - Report the new value of the configuration integrity control parameter back to the INITIATOR UE, via the CF parameter in the *DefineGroupResp* service primitive.
- The RESPONDER UE shall return "OK" status (parameter *Result(I)* = *result-ok* for all Is) if and only if the configuration integrity control parameter has been updated OK.

⁵⁸⁾ Though the protocol is not limiting the size of data, the implementation may have to. Any implementation should at least be able to respond with data according to NIST SP 500-177 (see bibliography [1]).

⁵⁹⁾ This index is not to be interpreted as an index into an array *Objid(I)* in the usual sense, but only as an integer ordering number for the elements coded into the *Objid* parameter. The elements (Object identifiers) are ASCII text strings, all consecutively coded into a one-dimensional array of octets; see IEC 60870-6-501, A.2.4. Increasing the value of "index I" must correspond to moving to a greater index value within this octets array.

⁶⁰⁾ See the APDU description, about the configuration integrity control parameter and associated functions.

- e) A réception d'une primitive de service *DefineGroupConf* valide, l'INITIATOR UE doit stocker la valeur du paramètre de contrôle de l'intégrité de la configuration reçue de la primitive parmi les données propres au RESPONDER UE en fonctionnement, dans le système INITIATOR⁶¹⁾.
 - f) Les règles indiquées en 6.1 s'appliquent également ici.
 - g) Les règles indiquées dans la procédure de mise à jour du Configuration Set, en 5.4, s'appliquent également.
- B) Si l'invocation a pour but une définition:

L'Object set d'un groupe peut normalement être codé dans un seul paramètre Objid. Le cas definition est alors intégré au cas redefinition: une seule transaction *DefineGroup*.

Si la définition par sous-groupes est utilisée, la séquence de la primitive est composée de transactions *DefineGroup* consécutives (voir A ci-dessus), chacune couvrant les plages contiguës suivantes de valeurs croissantes d'Object number:

Transaction n°	Objet n° borne inférieure	Objet n° borne supérieure
1	1	<max(1)> ⁶²⁾
2	<max(1)>+1	<max(2)>
–	–	–
–	–	–
j-1	<max(j-2)>+1	<max(j-1)>
j	<max(j-1)>+1	N
NOTE – N = <nombre total d'objets du groupe> – <max(j-1)>, N >= <max(j-1)>+1		

Règles:

- a) Les transactions individuelles doivent englober autant d'Object numbers consécutifs que l'autorise la TAPI (voir la note de bas de page pour la règle a) du cas redefinition ci-dessus), à l'exception de la dernière transaction de l'invocation de FU, qui englobe typiquement un jeu d'objets restant plus restreint.
- b) Les règles b) à g) du cas redefinition ci-dessus s'appliquent également ici pour chaque transaction.

⁶¹⁾ L'INITIATOR UE vérifiera par la suite sa valeur du paramètre de contrôle de l'intégrité de la configuration par rapport à la valeur acheminée par la primitive TASE Bind, lors des invocations ultérieures de *Permanent Association FU* ou de *Dynamic Association FU* transmises au même système RESPONDER. Cela a pour but de déterminer si la configuration du groupe est à considérer comme encore valide ou non: voir article 5.

⁶²⁾ La notation <max(i)> correspond au nombre total d'objets dont l'Object identifier a été mis à jour dans le CS(R) pendant l'invocation de FU en cours, à l'issue de la transaction n° i.

- e) Upon reception of a valid *DefineGroupConf* service primitive, the INITIATOR UE shall store the value of the configuration integrity control parameter received from the primitive, as part of the data specific for the current RESPONDER UE, in the INITIATOR system⁶¹⁾.
- f) The rules stated in 6.1 also apply here.
- g) The rules stated in the procedure for Configuration Set update, in 5.4, also apply.

B) If invoked for definition:

Normally, the Object set of a group can be coded into a single Objid parameter. The definition case is then collapsed into the redefinition case: one single *Define Group* transaction.

If definition by subgroups is used, the primitive sequence is composed of consecutive *Define Group* transactions (see A, above), each covering the following contiguous ranges of ascending Object number values:

Transaction No.	Object No. lower bound	Object No. upper bound
1	1	<max(1)> ⁶²⁾
2	<max(1)> + 1	<max(2)>
–	–	–
–	–	–
j – 1	<max(j – 2)> + 1	<max(j – 1)>
j	<max(j – 1)> + 1	N
NOTE – N = <total number of objects in group> – <max(j – 1)>, N >= <max(j – 1)> + 1		

Rules:

- a) The individual transactions shall encompass as many consecutive Object numbers as allowed by the TAPI (see footnote at rule a) for the redefinition case, above), except for the last transaction of the FU invocation, which typically will encompass a smaller remaining set of objects.
- b) Rules b) through g) for the redefinition case, above, also apply here, for each individual transaction.

⁶¹⁾ The INITIATOR UE will later check this value of the configuration integrity control parameter against the value conveyed by the TASE Bind primitive, during each subsequent *Permanent Association FU* or *Dynamic Association FU* invocation to the same RESPONDER system. This is done in order to determine whether the group configuration is still to be considered valid or not: see clause 5.

⁶²⁾ Notation: <max(i)> is the accumulated number of objects whose Object identifier has been updated in the CS(R) during the current FU invocation, at the completion of transaction No. i.

6.3.3.1.2 Parameter values

DefineGroup:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Group type	Valeur <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> , <i>Binary-command-group</i> , <i>Analog-setpoint-group</i> , <i>Digital-setpoint-group</i> ou <i>Text-message-group</i> . La valeur doit être égale à la valeur donnée de l'attribut Group type du CS(R) pour le n° de groupe Group-number.	Copie de la valeur à partir de <i>Ind</i>
Group number	Numéro de référence pour le groupe en question. Valeur de l'attribut Group number dans le jeu d'attributs du CS(R), pour laquelle l'attribut Object set est à modifier.	Copie de la valeur à partir de <i>Ind</i>
Index1	Borne inférieure de la plage de valeurs contiguës de l'attribut Object number pour lequel la valeur correspondante de l'attribut Object identifier est à mettre à jour, dans le CS(R). Doit être ≥ 1 et $\leq$ à la valeur donnée de l'attribut Group size du groupe, dans le CS(R). Si l'Object set entier du groupe est contenu dans le paramètre Objid, doit être = 1.	Copie de la valeur à partir de <i>Ind</i>
Index2	Borne supérieure de la plage de valeurs contiguës de l'attribut Object number pour lequel la valeur correspondante de l'attribut Object identifier est à mettre à jour, dans le CS(R). Doit être $\geq$ Index1 et $\leq$ à la valeur donnée de l'attribut Group size du groupe, dans le CS(R). Si l'Object set entier du groupe est contenu dans le paramètre Objid, doit être égal au nombre d'Object identifiers codés dans Objid.	Copie de la valeur à partir de <i>Ind</i>
CF	(Non applicable)	Valeur du Control Field pour l'INITIATOR UE qui a invoqué la <i>Group Definition FU</i> . Voir 5.4.
Objid	Jeu des valeurs de l'attribut Object identifier du groupe. Les valeurs sont à placer aux index croissants consécutifs (I) ⁶³⁾ par ordre croissant consécutif de la valeur de leur attribut Object number du CS(I). Doit contenir le nombre de valeurs suivant: $(\text{Index2} - \text{Index1}) + 1$. Les règles de codage se trouvent en A.2.4.1 de la CEI 60870-6-501. Pour la security class 2: le code d'authentification est ajouté après l'octet zéro de fin. Le code d'authentification est généré sur la base du paramètre Objid(I), en excluant l'octet zéro de fin. La taille de l'Objid(I) plus le code d'authentification ne doit pas dépasser la taille maximale. Pour la security class 3: le paramètre Objid(I) est crypté, à l'exclusion de l'octet zéro de fin. Une somme de contrôle doit être présente. Diverses manières de procéder sont décrites en 9.5.	(Non applicable)

⁶³⁾ Voir les notes 59 et 60.

6.3.3.1.2 Parameter values

Define Group:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Group type	Value <i>Measurand-group, Status-group, Discrete-group, Logical-breaker-status-group, Binary-command-group, Analog-setpoint-group, Digital-setpoint-group</i> or <i>Text-message-group</i> . Value shall be equal to assumed value of attribute Group type in the CS(R) for group No. Group Number.	Copy of value from <i>Ind</i>
Group-number	Reference number for the group in question. Value of attribute Group number in set of attributes in the CS(R), for which attribute Object set is to be modified.	Copy of value from <i>Ind</i>
Index1	Lower bound of contiguous range of values of component attribute Object number for which the corresponding value of component attribute Object identifier is to be updated, in the CS(R). Shall be ≥ 1 and $\leq$ assumed value of attribute Group size for the group, in the CS(R). If whole Object set of group is contained within the Objid parameter: shall be = 1.	Copy of value from <i>Ind</i>
Index2	Upper bound of contiguous range of values of component attribute Object number for which the corresponding value of component attribute Object identifier is to be updated, in the CS(R). Shall be $\geq$ Index1 and $\leq$ assumed value of attribute Group size for the group, in the CS(R). If whole Object set of group is contained within the Objid parameter: shall be equal to the number of Object identifiers coded into Objid.	Copy of value from <i>Ind</i>
CF	(Not applicable)	Control Field value for the INITIATOR UE that invoked the <i>Group Definition FU</i> . See 5.4.
Objid(l)	Set of values of component attributes Object identifier for the group. The values are to be placed at ascending consecutive indices (l) ⁶³⁾ , in ascending consecutive order of value of their attribute Object number from the CS(l). Shall contain the following number of values: $(\text{Index2} - \text{Index1}) + 1$. Coding rules are to be found in IEC 60870-6-501, A.2.4.1. For security class 2: the authentication code is appended after the terminating zero octet. The authentication code is generated on the basis of the Objid(l) parameter excluding the terminating zero octet. The size of the Objid(l) plus the authentication code shall not exceed the maximum size. For security class 3: the Objid(l) parameter excluding the terminating zero octet is enciphered. A checksum shall be present. Possible ways to do this are described in 9.5.	(Not applicable)

⁶³⁾ See footnotes 59) and 60).

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Result(I)	(Non applicable)	Un code d'erreur sous forme de nombre entier par objet dans le paramètre Objid. La valeur la plus faible d'index I, est 1. Le code à l'index I concerne l'objet codé dans le paramètre Objid à l'emplacement du numéro de référence I ⁶⁴⁾ Il est admis que Result(I) contienne à un moment donné plusieurs codes d'erreur (erreurs simultanées pour plus d'un objet). Action effectuée comme spécifié: Valeur du premier élément (I=1): <i>result-ok</i> Action non effectuée, du fait d'une condition d'erreur: Si l'erreur a pour résultat un code d'erreur <i>gtype-out-of-range</i>, <i>gnr-out-of-range</i>, <i>configuration-buffer-overflow</i>, <i>not-reconfigurable</i> ou <i>index-out-of-range</i>: Valeur du premier élément (I=1): Le code d'erreur. Si l'erreur a pour résultat un code d'erreur <i>objlength-out-of-range</i> ou <i>object-identifie-unknown</i> pour l'objet correspondant à l'élément n° I: Valeur de l'élément n° I: Le code d'erreur. Noter que la transaction entière doit échouer dans le RESPONDER UE (pas de mise à jour du CS(R)) si elle échoue pour au moins un objet. Le RESPONDER UE est autorisé à choisir de toujours coder un élément par objet défini dans le paramètre Result(I), quel que soit le degré de succès, ou, sinon, à terminer le paramètre Result(I) après la valeur I correspondant au plus grand numéro d'objet en erreur. Voir aussi 6.3.3.2. Les règles de codage se trouvent en A.2.4.3 de la CEI 60870-6-501.

6.3.3.2 Traitement des erreurs

6.3.3.2.1 FU disruption

Interruption par la *Permanent Association FU* ou la *Dynamic Association FU*:

L'interruption de la partie INITIATOR et de la partie RESPONDER de l'invocation en cours de la *Group Definition FU* doit être déclenchée localement, dans le cadre du traitement des primitives *AbortInd* entrantes de la partie INITIATOR et de la partie RESPONDER de l'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* qui traite l'association où l'invocation de la *Group Definition FU* est en fonctionnement.

Les deux parties de l'invocation en cours de la *Group Definition FU* doivent être terminées correctement, sans que l'une ou l'autre des parties n'essaie d'émettre une primitive associée à la fin elle-même.⁶⁵⁾

⁶⁴⁾ Voir les notes 59 et 60.

⁶⁵⁾ Les modes opératoires locaux de nettoyage ne sont pas spécifiés dans le présent rapport technique.

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Result(I)	(Not applicable)	One integer error code per object in the Objid parameter. Lowest value of index, I, is 1. The code at index I concerns the object coded into parameter Objid at location reference number⁶⁴). Result(I) may contain more than one error code at a time (simultaneous errors for more than one object). Action performed as specified: Value of first element (I=1): result-ok. Action not performed, due to error condition: If error resulting in error code <i>gtype-out-of-range</i>, <i>gnr-out-of-range</i>, <i>configuration-buffer-overflow</i>, <i>not-reconfigurable</i> or <i>index-out-of-range</i>: Value of first element (I=1): The error code. If error resulting in error code <i>Objlength-out-of-range</i> or <i>object-identifier-unknown</i> for the object corresponding to element No. I: Value of element No. I: The error code. Note that the whole transaction shall fail in the RESPONDER UE (no update of the CS(R)), if it fails for at least one object. The RESPONDER UE may choose to always code one element per defined object into the Result(I) parameter, irrespective of the degree of success, or alternatively, to terminate the Result(I) parameter after the I value corresponding to the greatest object number which is in error. Also see 6.3.3.2. Coding rules are to be found in A.2.4.3 of IEC 60870-6-501.

6.3.3.2 Error handling

6.3.3.2.1 FU disruption

Disruption by the *Permanent Association FU* or *Dynamic Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Group Definition FU* shall be triggered locally, as a part of the handling of incoming *AbortInd* primitives in both the INITIATOR part and the RESPONDER part of the *Permanent Association FU* or *Dynamic Association FU* invocation handling the association where the *Group Definition FU* is running.

Both parts of the current invocation of the *Group Definition FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself⁶⁵).

⁶⁴) See footnotes 59) and 60).

⁶⁵) Local clean-up procedures are not specified by this technical report.

6.3.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Group Definition FU* n'est pas présente dans un INITIATOR UE:

Le traitement de ce type d'erreur résultant d'une tentative d'invocation par la Coordinating Function locale constitue un problème local qui ne fait pas partie du domaine d'application de ce rapport.

Si ce type d'erreur survient lors d'une tentative d'invocation du fait d'une invocation de la *Group Configuration FU*, l'erreur est traitée par le mécanisme de traitement des erreurs de la *Group Configuration FU*. Voir la description de la *Group Configuration FU*.

Si la *Group Definition FU* n'est pas présente dans un RESPONDER UE:

Le RESPONDER UE doit répondre aux tentatives d'invocation de la manière suivante:

- émettre une primitive *DefineGroupResp* avec Result = *remote-service-user-unavailable*.

FU présente, mais tentative illégale:

Les tentatives d'invocation qui violent les règles indiquées en 6.3.2.3.2 dans un INITIATOR UE constituent un problème local qui ne fait pas partie du domaine d'application du présent rapport technique.

Dans un RESPONDER UE, les tentatives d'invocation qui violent les mêmes règles doivent être traitées par le RESPONDER UE de la manière suivante:

- émission d'une primitive *DefineGroupResp* avec Result = *remote-service-user-unavailable*, sans (ré)invocation de la *Group Definition FU* pour le groupe concerné, dans le RESPONDER UE.

6.3.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>DefineGroupConf</i>
FU non en fonctionnement	Ignorer, ou Indication d'erreur locale
FU en fonctionnement, en attente de <i>DefineGroupConf</i>	(Normal)

Partie RESPONDER: Non applicable: une primitive de service *DefineGroupInd* n'est jamais hors contexte dans le RESPONDER.⁶⁶⁾

6.3.3.2.4 Timing errors

Erreur dans la partie INITIATOR: Non applicable. Aucune contrainte de synchronisation n'est imposée à la partie INITIATOR de la *Group Definition FU*.

⁶⁶⁾ L'ordonnement correct de la primitive *Ind/Resp* est imposé par le TASE.1. Le présent rapport technique ne spécifie pas l'imposition de l'ordonnement des sous-groupes par des transactions *Define Group* individuelles, de la part du RESPONDER UE. Au contraire, le RESPONDER UE doit accepter les définitions de sous-groupes entrantes dans n'importe quel ordre.

6.3.3.2.2 Illegal invocation attempt

FU not present:

If the *Group Definition FU* is not present in an INITIATOR UE:

The handling of this type of error resulting from an invocation attempt by the local Coordinating Function is a local issue, outside the scope of this report.

If this type of error occurs upon an invocation attempt by a *Group Configuration FU* invocation, the error is handled by the error handling mechanism within the *Group Configuration FU*. See the *Group Configuration FU* description.

If the *Group Definition FU* is not present in a RESPONDER UE:

The RESPONDER UE shall respond to activation attempts in the following way:

- Issuing a *DefineGroupResp* primitive with Result = *remote-service-user-unavailable*.

FU present, but attempt illegal:

Invocation attempts violating the rules stated in 6.3.2.3.2 in an INITIATOR UE is a local issue, outside the scope of this technical report.

In a RESPONDER UE, invocation attempts violating the same rules shall be handled by the RESPONDER UE in the following way:

- Issue a *DefineGroupResp* primitive with Result = *remote-service-user-unavailable* without (re-)invoking the *Group Definition FU* for the group concerned, in the RESPONDER UE.

6.3.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>DefineGroupConf</i>
FU not running	Ignore, or Local error indication
FU running, waiting for <i>DefineGroupConf</i>	(Normal)

RESPONDER part: Not applicable: A *DefineGroupInd* service primitive is never out of context in the RESPONDER⁶⁶⁾.

6.3.3.2.4 Timing errors

Error in INITIATOR part: Not applicable: There are no timing restraints placed upon the INITIATOR part of the *Group Definition FU*.

⁶⁶⁾ Proper *Ind/Resp* primitive sequencing is enforced by the TASE.1. This technical report does not specify any enforcement of subgroup sequencing across individual Define Group transactions, on the part of the RESPONDER UE. On the contrary, the RESPONDER UE shall accept incoming subgroup definitions in any order.

Erreur dans la partie RESPONDER:

Erreur	Réaction de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à <i>DefineGroupInd</i>	Sur le RESPONDER: Erreur locale lors d'une éventuelle tentative d'émission de <i>DefineGroupResp</i>	Partie RESPONDER: Terminer l'invocation de la FU localement.
	Sur l'INITIATOR: <i>DefineGroupConf</i> , avec Result = <i>remote-service-user-unavailable</i> .	Partie INITIATOR: Terminer l'invocation de la FU localement.

6.3.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsqu'elle se produit lors d'une tentative de *DefineGroupReq*:

- terminer l'invocation de la FU localement.

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsqu'elle se produit lors d'une tentative de *DefineGroupResp*:

- terminer l'invocation de la FU localement (la partie INITIATOR de la FU sera finalement terminée après expiration du délai dans le fournisseur de TASE.1).

Voir 4.2.1.

6.3.3.2.6 TASE.1 service primitive parameters errors

Erreurs dans la primitive *DefineGroupInd* (détectées par la partie RESPONDER):

A) Erreurs liées au groupe:

La partie RESPONDER doit dans tous les cas de ce type émettre une *DefineGroupResp* avec Result(1)⁶⁷⁾ = <Valeur du tableau ci-dessous>, et les valeurs de Group-number, Group-type, Index1 et Index2 comme dans le *Ind* correspondant. Le nombre d'éléments de Result doit être ≥ 1 . Si une erreur liée au groupe est détectée, Result(i), $i > 1$, doit avoir la valeur *result-ok*, s'il est présent.

⁶⁷⁾ Première valeur du tableau Result.

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>DefineGroupInd</i>	In RESPONDER: Local error from eventual attempt in issuing <i>DefineGroupResp</i> In INITIATOR: <i>DefineGroupConf</i> , with Result = <i>remote-service-user-unavailable</i> .	RESPONDER part: Terminate FU invocation locally. INITIATOR part: Terminate FU invocation locally.

6.3.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with a *DefineGroupReq* attempt:

- Terminate FU invocation locally.

See 4.2.1.

RESPONDER part of the FU:

When occurring with a *DefineGroupResp* attempt:

- Terminate FU invocation locally. (INITIATOR part of FU will eventually be terminated after timeout in the TASE.1 provider).

See 4.2.1.

6.3.3.2.6 TASE.1 service primitive parameter errors

Errors in the *DefineGroupInd* primitive (detected by the RESPONDER part):

A) Group-related errors:

The RESPONDER part shall in all these cases issue *DefineGroupResp* with Result(1)⁶⁷⁾ = <Value from table below>, and the values of Group-number, Group-type, Index1 and Index2 as in the corresponding *Ind*. The number of elements in Result shall be ≥ 1. If a group-related error is detected, Result(i), i>1, shall have the value *result-ok*, if present.

⁶⁷⁾ The first value in the array Result.

La partie RESPONDER de la FU ne doit pas être invoquée.

Erreur	Valeur du paramètre Result(1)
Valeur de Group-type non égale à la valeur de l'attribut Group type du CS(R) pour le n° de groupe Group-number. (Le cas du n° de groupe Group-number non existant est étudié ci-dessous.)	<i>gtype-out-of-range</i>
Valeur de Group-number illégale ou n° de groupe Group-number non existant dans le CS(R) ou non créé par le biais de l'interface TASE.1.	<i>gnr-out-of-range</i>
Index1 <= 0 ou Index2 <= 0	<i>index-out-of-range</i>
Index1 en dehors de la plage légale des valeurs de l'attribut Object number pour les objets définis dans le groupe n° Group-number du CS(R). Plage légale: 1 – Group size.	<i>index-out-of-range</i>
Index2 < Index1 ou Index2 > borne supérieure des valeurs de l'attribut Object number pour les objets définis du groupe n° Group-number du CS(R). Borne supérieure: Group size.	<i>index-out-of-range</i>
Nombre d'Object identifiants dans Objid en dehors de la plage légale des valeurs de l'attribut Object number pour le groupe n° Group-number du CS(R). Plage légale: 1 – Group size.	<i>index-out-of-range</i>
Nombre d'Object identifiants dans Objid non égal à Index2 – Index1 + 1	<i>index-out-of-range</i>
Les <i>DefineGroupInd</i> consécutifs doivent contenir des Index consécutifs (Index1 dans le n ^{ième} <i>DefineGroupInd</i> > Index2 dans le (n-1) ^{ième} <i>DefineGroupInd</i>).	<i>index-out-of-range</i>
Il n'est pas possible de mettre à jour l'Object set du groupe n° Group-number du CS(R) comme spécifié à cause des limitations des ressources locales.	<i>configuration-buffer-overflow</i>
Attribut Persistent = <i>true</i> pour le groupe n° Group-number du CS(R).	<i>not-reconfigurable</i>
Attribut Static = <i>true</i> et au moins un Object identifier défini pour le groupe n° Group-number du CS(R). ⁶⁸⁾	<i>not-reconfigurable</i>
RESPONDER UE pas en mesure d'exécuter la mise à jour requise du CS(R) pour une raison autre que celles énumérées ci-dessus et que celles du tableau de B) ci-dessous.	<i>remote-service-user-unavailable</i>
Pour la security class 2: le code d'authentification reçu est >< au code d'authentification généré basé sur les données reçues.	<i>invalid-authentication-code-received</i>
Pour la security class 3: la somme de contrôle reçue est >< à la somme de contrôle générée pendant le décryptage.	<i>decipherment-error</i>

Les erreurs liées au groupe doivent prévaloir sur les erreurs liées aux objets (ci-dessous): les erreurs liées aux objets ne doivent être signalées que si aucune erreur liée au groupe n'est détectée.

B) Erreurs liées aux objets:

Dans tous ces cas, la partie RESPONDER doit émettre *DefineGroupResp* avec les valeurs Group-number, Group-type, Index1 et Index2 comme dans le *Ind* correspondant. Result(i)⁶⁹⁾ doit contenir:

- Pour tous les index (I) correspondant aux objets pour lesquels une erreur a été détectée:
<Valeur du tableau ci-dessous>
- Pour tous les index (I) correspondant aux objets pour lesquels aucune erreur n'a été détectée:
Valeur *result-ok*.

⁶⁸⁾ Noter que si aucun Object identifier n'est défini (groupe vide), les transactions *DefineGroup* sont toutefois légales.

⁶⁹⁾ La valeur du tableau Result pour la valeur d'index i est égale au nombre entier d'ordonnement de l'Object Identifier du paramètre Objid (les nombres entiers d'ordonnement Objid sont décrits dans les notes 59 et 60.

The RESPONDER part of the FU must not be invoked.

Error	Value of parameter Result (1)
Group-type value not equal to value of attribute Group type in CS(R) for group No. Group-number. (The case of non-existing group No. Group-number is considered below.)	<i>gtype-out-of-range</i>
Value of Group-number is illegal, or group No. Group-number does not exist in the CS(R) or was not created via the TASE.1 interface	<i>gnr-out-of-range</i>
Index1 <= 0 or Index2 <= 0	<i>index-out-of-range</i>
Index1 outside legal range of values of component attribute Object number for defined objects in group No. Group-number in CS(R). Legal range: 1 – Group size.	<i>index-out-of-range</i>
Index2 < Index1 or Index2 > upper bound of values of component attribute Object number for defined objects in group No. Group-number in CS(R). Upper bound: Group size.	<i>index-out-of-range</i>
Number of Object identifiers in Objid outside legal range of values of component attribute Object number for group No. Group-number in CS(R). Legal range: 1 – Group size.	<i>index-out-of-range</i>
Number of Object identifiers in Objid not equal to Index2 – Index1 + 1.	<i>index-out-of-range</i>
Consecutive <i>DefineGroupInd</i> must contain consecutive Indices (Index1 in the <i>n</i> th <i>DefineGroupInd</i> > Index2 in the (<i>n</i> – 1)th <i>DefineGroupInd</i>).	<i>index-out-of-range</i>
The Object set of group No. Group-number in CS(R) cannot be updated as specified, because of local resource limitations	<i>configuration-buffer-overflow</i>
Attribute Persistent = <i>true</i> , for group No. Group-number in CS(R).	<i>not-reconfigurable</i>
Attribute Static = <i>true</i> , and at least one Object identifier defined, for group No. Group-number in CS(R) ⁶⁸⁾	<i>not-reconfigurable</i>
RESPONDER UE not able to perform the requested CS(R) update, due to any reason other than those listed above, and those in the table in B, below.	<i>remote-service-user-unavailable</i>
For security class 2: The received authentication code >< the generated authentication code based on the received data.	<i>invalid-authentication-code-received</i>
The security class 3: The received checksum >< the generated checksum during the decipherment.	<i>decipherment-error</i>

Group-related errors shall have precedence over object-related errors (below): Object-related errors shall not be reported unless no group-related error is detected.

B) Object-related errors:

The RESPONDER part shall in all these cases issue *DefineGroupResp* with the values of Group-number, Group-type, Index1 and Index2 as in the corresponding *Ind*. Result(i)⁶⁹⁾ shall contain:

- At all indices(i) corresponding to objects for which an error has been detected:
 <Value from table below>
- At all indices(i) corresponding to objects for which no error has been detected:
 Value *result-ok*.

⁶⁸⁾ Note that if no Object identifier is defined (empty group), *Define Group* transactions are however legal.

⁶⁹⁾ The value of the array Result at index value i equal to the integer ordering number of the corresponding Object identifier of the Objid parameter. (Integer ordering numbers in Objid are described in footnotes 59) and 60))

Le nombre d'éléments de Result (valeur maximale de index i) doit être suffisant pour englober tous les index correspondant aux objets pour lesquels une erreur a été détectée. Il n'est pas nécessaire que la dernière valeur de Result (pour i max.) soit différente de *result-ok*.⁷⁰⁾

La partie RESPONDER de la FU ne doit pas être invoquée.

Erreur	Valeur du paramètre Result(i)
Nombre d'octets d'Object identifier pour l'objet n° i dans Objid supérieur à l'attribut Object identifier size pour le groupe n° Group-number du CS(R).	<i>objlength-out-of-range</i>
Erreur de format dans Objid, rendant le décodage d'Object identifier impossible pour l'objet n° i.	<i>objlength-out-of-range</i>
Valeur d'Object identifier pour l'objet n° i dans Objid qui n'est pas un identificateur légal pour de tels objets dans le système RESPONDER.	<i>objid-unknown</i>

Erreurs dans la primitive *DefineGroupConf* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Absence de correspondance entre Group-type/Index1/Index2 dans la primitive et Group-type/Index1/Index2 dans la <i>DefineGroupReq</i> correspondante, et Result(i) = <i>result-ok</i> pour tous les i.	Terminer l'invocation de la FU localement , en enregistrant l'erreur. Mettre à jour le paramètre de contrôle d'intégrité de la configuration de manière normale ⁷¹⁾
Result(i) >< <i>result-ok</i> pour un i au moins.	Terminer l'invocation de la FU localement , en enregistrant l'erreur. Ne pas mettre à jour le paramètre de contrôle d'intégrité de la configuration.

6.4 Group Readout FU

Type: Primary.

6.4.1 Fonction

Une invocation de la *Group Readout FU* permet à l'INITIATOR UE qui exécute l'invocation d'extraire dans sa totalité le jeu d'attributs dont fait partie une valeur spécifiée de Group number, depuis le CS(R) du système RESPONDER. Ceci équivaut à obtenir un affichage d'une configuration de groupe dans le CS(R).

Cette fonction ne doit pas être limitée aux groupes qui sont configurés par le biais de TASE.1: les groupes configurés dans le CS(R) par tout autre système approprié doivent également être inclus.

Noter que le fonctionnement de la *Group Management FU* et de la *Group Definition FU*, au contraire, est limité à la valeur des paramètres Persist et Static.

Les parties INITIATOR et RESPONDER de cette FU sont généralement asymétriques par rapport aux événements d'invocation et de fin:

⁷⁰⁾ Ce qui signifie que le RESPONDER UE est autorisé à choisir de toujours émettre un paramètre Result(i) complet: une valeur par objet, quel que soit le degré de succès de la transaction.

⁷¹⁾ Etant donné que Result = *result-ok*, le paramètre de contrôle de l'intégrité de la configuration aura été mis à jour dans le système RESPONDER.

The number of elements in Result (maximum value of index *i*) shall be sufficient to encompass all indices corresponding to objects for which an error has been detected. The last value in Result (at max. *i*) is not required to be different from *result-ok*.⁷⁰⁾

The RESPONDER part of the FU shall not be invoked.

Error	Value of parameter Result(<i>i</i>)
Number of octets in Object identifier for object No. <i>i</i> in Objid greater than attribute Object identifier size for group No. Group-number in CS(R)	<i>objlength-out-of-range</i>
Format error in Objid, making decoding of Object identifier for object No. <i>i</i> impossible	<i>objlength-out-of-range</i>
Value of Object identifier for object No. <i>i</i> in Objid not a legal identifier for such objects in the RESPONDER system	<i>objid-unknown</i>

Errors in the *DefineGroupConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Mismatch between Group-type/Index1/Index2 in primitive and Group-type/Index1/Index2 in corresponding <i>DefineGroupReq</i> , and Result(<i>i</i>) = <i>result-ok</i> for all <i>i</i> .	Terminate the FU invocation locally, registering the error. Update the configuration integrity control parameter, as normal. ⁷¹⁾
Result(<i>i</i>) >< <i>result-ok</i> for at least one <i>i</i> .	Terminate the FU invocation locally, registering the error. Do not update the configuration integrity control parameter.

6.4 Group Readout FU

Type: Primary.

6.4.1 Function

A *Group Readout FU* invocation allows the invoking INITIATOR UE to fetch the complete attribute set, of which a specified value of Group number is part, from the CS(R) in the RESPONDER system. This is equivalent to obtaining a readout of a group configuration in the CS(R).

The function shall not be limited to groups that are configured via the TASE.1: groups configured in the CS(R) by any relevant system shall also be included.

Note that the operation of both the *Group Management FU* and the *Group Definition FU*, on the other hand, is restricted to the value of the parameters Persist and Static.

The INITIATOR and RESPONDER parts of this FU are generally asymmetrical with regard to invocation and termination events:

⁷⁰⁾ Meaning that the RESPONDER UE may choose to always send a full-sized Result(*i*) parameter: one value per object, irrespective of the degree of transaction success.

⁷¹⁾ Because Result = *result-ok*, the configuration integrity control parameter will have been updated in the RESPONDER system.

La partie RESPONDER n'a aucun moyen de savoir si une transaction *GetGroup* terminée est la dernière d'une série de transactions constituant l'invocation de la *Group Readout FU*; voir 6.4.3.1.1. En conséquence, la partie RESPONDER est généralement invoquée un certain nombre de fois, une fois pour chaque transaction *GetGroup* entrante. La partie INITIATOR, au contraire, n'est invoquée qu'une seule fois par l'invocation de la *Group Readout FU* proprement dite. Il convient donc de comprendre l'invocation de la *Group Readout FU* comme signifiant strictement «invocation de la partie INITIATOR de la *Group Readout FU*». C'est toutefois la formulation «invocation de la *Group Readout FU*» qui est utilisée tout au long de ce rapport.

6.4.2 Règles de coordination

6.4.2.1 Utilisation d'une association

Toutes les interactions TASE.1 faisant partie d'une invocation de la *Group Readout FU* sont véhiculées par une seule association. L'association doit posséder les caractéristiques spécifiées en 6.4.2.3.1.

6.4.2.2 Relations avec les autres FU

6.4.2.2.1 Invoking FUs

La *Group Readout FU* ne doit être invoquée par aucune autre FU.

6.4.2.2.2 Invoked FUs

La *Group Readout FU* ne doit invoquer aucune autre FU.

6.4.2.2.3 Disrupting FUs

Il est admis d'interrompre la *Group Readout FU* par:

- la *Permanent Association FU*;
- la *Dynamic Association FU*.

6.4.2.2.4 Disrupted FUs

La *Group Readout FU* ne doit interrompre aucune autre FU.

6.4.2.3 Invocation

6.4.2.3.1 Conditions préalables

L'une⁷²⁾ des FU suivantes doit avoir été invoquée avant toute invocation de la *Group Readout FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*.

L'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* doit encore être en fonctionnement au moment de l'invocation de la *Group Readout FU*. Dans le cas de la *Permanent Association FU*, l'association maintenue par l'invocation doit être en fonctionnement (et non interrompue temporairement) à ce moment-là.

La *Permanent Association FU* ou la *Dynamic Association FU* doivent avoir été invoquées afin de créer (et, pour la *Permanent Association FU*, également de maintenir) l'association à utiliser pour les interactions liées à l'invocation en cours de la *Group Readout FU*, avec la paire A-suffix spécifiée par le tableau de 5.2.1.

⁷²⁾ Sous réserve d'une décision locale dans l'INITIATOR UE.

The RESPONDER part has no way of knowing whether a completed *GetGroup* transaction is the last in the series of transactions constituting the *Group Readout FU* invocation; see 6.4.3.1.1. Consequently, the RESPONDER part is generally invoked a number of times, once for each incoming *GetGroup* transaction. The INITIATOR part, on the other hand, is invoked only once per invocation of the *Group Readout FU* itself. The term "*Group Readout FU* invocation" should therefore strictly read "INITIATOR part of *Group Readout FU* invocation". However, "*Group Readout FU* invocation" is used throughout this report.

6.4.2 Coordination rules

6.4.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Group Readout FU* are conveyed by one single association. The association shall have the characteristics as specified in 6.4.2.3.1.

6.4.2.2 Relation to other FUs

6.4.2.2.1 Invoking FUs

The *Group Readout FU* shall not be invoked by any other FU.

6.4.2.2.2 Invoked FUs

The *Group Readout FU* shall not invoke any other FU.

6.4.2.2.3 Disrupting FUs

The *Group Readout FU* may be disrupted by:

- *Permanent Association FU*;
- *Dynamic Association FU*.

6.4.2.2.4 Disrupted FUs.

The *Group Readout FU* shall not disrupt any other FU.

6.4.2.3 Invocation

6.4.2.3.1 Prerequisites

One⁷²⁾ of the following FUs shall have been invoked preceding any invocation of the *Group Readout FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*.

The *Permanent Association FU* or *Dynamic Association FU* invocation shall still be running at the time of invocation of the *Group Readout FU*. In the case of *Permanent Association FU* the association maintained by the invocation shall be running (not temporarily broken) at that time.

The *Permanent Association FU* or *Dynamic Association FU* shall have been invoked in order to create (and, for the *Permanent Association FU*, also to maintain) the association to be used for the interactions related to the current invocation of the *Group Readout FU*, with the A-suffix pair specified in table by the table in 5.2.1.

⁷²⁾ Subject to local decision in the INITIATOR UE.

6.4.2.3.2 Restrictions

Aucune restriction n'est imposée à l'invocation de la *Group Readout FU*.

Toutefois, si la *Group Readout FU* est invoquée alors que la *Group Management FU* ou la *Group Definition FU* est en fonctionnement pour le même groupe, il se peut que la *Group Readout FU* échoue ou que l'INITIATOR UE reçoive des valeurs erronées. Cela dépend des détails d'implémentation de la base de données (non traités par le présent rapport technique), ainsi que de la synchronisation effective des invocations de FU.

6.4.2.3.3 Invoking events

Il est autorisé d'invoquer la partie INITIATOR de la *Group Readout FU* par:

- une demande locale par le biais de la Coordinating Function, dont l'origine est en dehors du domaine d'application du présent rapport technique.

L'invocation de la partie RESPONDER de la *Group Readout FU* est tentée chaque fois qu'une primitive *GetGroupInd* valide est reçue par le biais d'une association possédant les caractéristiques définies pour la *Group Readout FU*.

6.4.2.4 Fin

6.4.2.4.1 Orderly termination

Il est autorisé de déclencher la fin ordonnée de la partie INITIATOR d'une *invocation Group Readout FU* par:

- une erreur d'encombrement. Voir 6.4.3.2.5;
- la réception d'une primitive de service *GetGroupConf* non valide après l'émission d'une primitive de service *GetGroupReq* (fin suite à une erreur);
- la réception d'une primitive de service *GetGroupConf* valide après l'émission de la dernière primitive de service *GetGroupReq* de l'invocation en cours de la *Group Readout FU* (fin normale).

La partie RESPONDER d'une invocation de la *Group Readout FU* se termine toujours d'elle-même de manière ordonnée (fin normale) après avoir émis une primitive de service *GetGroupResp*.

6.4.2.4.2 Disruption

Il est autorisé d'interrompre les parties INITIATOR et RESPONDER d'une invocation *Group Readout FU* par:

- l'interruption d'une autre invocation de FU. Voir 6.4.2.2.3.

6.4.3 Procédures

6.4.3.1 Primitives de service TASE.1

Le service TASE.1 élémentaire suivant est utilisé par la *Group Readout FU*:

- *GetGroup*

6.4.2.3.2 Restrictions

There are no restrictions on invocation of the *Group Readout FU*.

However, if the *Group Readout FU* is invoked while the *Group Management FU* or the *Group Definition FU* is running for the same group, the *Group Readout FU* may fail, or the INITIATOR UE may receive erroneous values. This depends on database implementation details (not covered by this technical report), as well as actual timing of FU invocations.

6.4.2.3.3 Invoking events

The INITIATOR part of the *Group Readout FU* may be invoked by:

- Local request via the Coordinating Function, the original source of which is outside the scope of this technical report.

Invocation of the RESPONDER part of the *Group Readout FU* is attempted whenever a valid *GetGroupInd* primitive is received via an association with the characteristics as defined for the *Group Readout FU*.

6.4.2.4 Termination

6.4.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of a *Group Readout FU* invocation may be triggered by:

- congestion error. See 6.4.3.2.5;
- reception of an invalid *GetGroupConf* service primitive after issuing a *GetGroupReq* service primitive (termination on error);
- reception of a valid *GetGroupConf* service primitive after issuing the last *GetGroupReq* service primitive of the current *Group Readout FU* invocation (normal termination).

The RESPONDER part of a *Group Readout FU* invocation always terminates itself in an orderly manner (normal termination) after issuing a *GetGroupResp* service primitive.

6.4.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Group Readout FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 6.4.2.2.3.

6.4.3 Procedures

6.4.3.1 TASE.1 service primitives

The following elementary TASE.1 service is used by the *Group Readout FU*:

- *GetGroup*

6.4.3.1.1 Sequence

La séquence de base des primitives consiste en la seule transaction *GetGroup*:

INITIATOR UE	TASE.1	RESPONDER UE
<i>GetGroupReq</i>	----->	<i>GetGroupInd</i>
<i>GetGroupConf</i>	<-----	<i>GetGroupResp</i>

Normalement, l'affichage complet du groupe est traité pour une seule transaction *GetGroup*.

Du fait de restrictions d'implémentation, il est toutefois admis que la séquence de primitive complète soit composée d'un certain nombre de transactions successives *GetGroup* de base, conformément aux règles suivantes:

- a) Chaque transaction doit concerner un sous-groupe ou une plage de valeurs contiguës d'Object number. L'algorithme ci-dessous est obligatoire pour l'INITIATOR UE qui commande l'invocation de la *Group Readout FU*. N'ayant aucune connaissance préalable des valeurs d'attributs de groupe à l'exception de Group type et Group number, l'INITIATOR UE doit d'abord obtenir un maximum d'informations par le biais d'une seule transaction *GetGroup*. Un certain nombre de transactions supplémentaires doit suivre, en spécifiant toujours le nombre maximal restant d'objets et en enregistrant successivement les informations reçues, jusqu'à ce que:
- soit le nombre total maximal autorisé d'Object identifiants pour le groupe ait été renvoyé;
 - soit une transaction ne renvoie plus d'Object identifiants supplémentaires.

Algorithme pour INITIATOR UE dans la *Group Readout FU* (pseudocode informel) permettant de traiter les transactions *GetGroup* successives:

BEGIN

Exécuter une transaction Get Group, avec plage d'Object number 0 – 0 (paramètres Index1 et Index2 = 0).⁷³⁾

Enregistrer⁷⁴⁾, à partir de la transaction, les attributs de groupe suivants:

- Persistent (paramètre Persist)
- Static (paramètre Static)
- Priority (paramètre Priority)
- Group Size (paramètre MaxObj)
- Object identifier size (paramètre MaxObjSize)
- N paires de
 - Object number décompte successif des valeurs d'Object identifier extraites du paramètre Objid, en commençant par 1 pour la valeur à l'index le plus faible du paramètre Objid lorsque ce paramètre est considéré comme un tableau d'octets unidimensionnel, et en finissant par N, pour la dernière valeur extraite d'Objid)
 - Object identifier (paramètre Objid)

Fixer le décompte courant d'objet à N, comme déterminé à partir de la première transaction.

⁷³⁾ L'objet de cette transaction est d'établir une borne pour la LOOP ci-après, basée sur Group size, en extrayant en même temps autant d'Object identifiants que possible pendant la transaction. La plage Object number 0 – 0, qui est légale dans le contexte de ce FU, signifie «retourner autant d'Object identifiants que possible».

⁷⁴⁾ C'est-à-dire renvoyer à la Coordinating Function.

6.4.3.1.1 Sequence

The basic sequence of primitives is the single *GetGroup* transaction:

INITIATOR UE	TASE.1	RESPONDER UE
<i>GetGroupReq</i>	----->	<i>GetGroupInd</i>
<i>GetGroupConf</i>	<-----	<i>GetGroupResp</i>

Normally, the complete group readout is handled by one single *GetGroup* transaction.

Due to implementation restrictions, however, the complete primitive sequence may be composed of a number of successive basic *GetGroup* transactions, according to the following rules:

- a) Each transaction shall cover a subgroup, or contiguous range of Object number values. The algorithm below is mandatory for the INITIATOR UE controlling the *Group Readout FU* invocation. Not having any previous knowledge about group attribute values except for Group type and Group number, the INITIATOR UE shall first obtain the maximum amount of information via a single *GetGroup* transaction. Then there shall follow a number of additional transactions, always specifying the maximum possible remaining number of objects and successively registering the received information, until either:
 - the maximum allowed accumulated number of Object identifiers for the group has been returned;
 - a transaction returns no further Object identifiers.

Algorithm for INITIATOR UE in *Group Readout FU* (informal pseudocode) to handle successive *GetGroup* transactions:

BEGIN

Perform a *GetGroup* transaction, with Object number range 0 – 0 (parameters Index1 and Index2 both =0).⁷³⁾

Register⁷⁴⁾, from transaction, the following group attributes:

- Persistent (parameter Persist)
- Static (parameter Static)
- Priority (parameter Priority)
- Group Size (parameter MaxObj)
- Object identifier size (parameter MaxObjSize)
- N pairs of
 - Object number (successive count of Object identifier values fetched from parameter Objid, starting with 1 for the value at the lowest index of the Objid parameter when that parameter is viewed as a one-dimensional array of octets, and ending with N, for the last value from Objid)
 - Object identifier (parameter Objid)

Set current object count = N, as determined from the first transaction.

⁷³⁾ The purpose of this transaction is to establish a bound for the LOOP below, based on Group size, at the same time extracting as many Object identifiers as possible during the transaction. The semantics of the Object number range 0 – 0, which is legal within the context of this FU, is "return as many Object identifiers as possible".

⁷⁴⁾ That is, report to the Coordinating Function.

LOOP:

IF le décompte courant des objets est inférieur à la valeur de Group size, comme déterminé à partir de la première transaction:

Exécuter une nouvelle transaction, avec:

Index1 = décompte d'objets courant, plus un

Index2 = valeur de Group size, comme déterminé à partir de la première transaction

IF le paramètre Objid n'est pas vide (s'il contient au moins un Object identifier)⁷⁵⁾

Enregistrer, à partir de la transaction, l'attribut Object identifier pour chaque objet, comme décodé à partir du paramètre Objid.⁷⁶⁾

Mettre à jour le décompte d'objets courant, en ajoutant le nouveau nombre d'objets décodés à partir du paramètre Objid à la valeur précédente du décompte d'objets courant.

ELSE

Exit LOOP

END-IF

ELSE

Exit LOOP

END-IF

END-LOOP

Terminer l'invocation de FU.

END

Dans ce cas, noter que le nombre de transactions requis pour obtenir un affichage complet du groupe est normalement supérieur de un au nombre de transactions requis pour transporter le jeu réel d'Object identifiers: la dernière transaction contient habituellement zéro Object identifiers⁷⁷⁾.

Ceci s'applique également aux groupes pour lesquels la valeur de l'attribut Group size n'est pas supérieure à la valeur qui permet à la PDU *GetGroupConf* de contenir la configuration complète, et aux groupes pour lesquels la valeur de l'attribut Group size est supérieure à la valeur qui permet à la PDU *GetGroupConf* de contenir la configuration complète mais où le nombre effectif d'Object identifiers est assez petit pour le permettre:

Dans les deux cas, l'invocation de la *Group Readout FU* englobera deux transactions *Get Group*.

La raison sous-jacente est qu'une transaction *GetGroup* n'a aucun moyen de signaler simultanément une configuration complète et le fait que la configuration est complète.

- b) Le RESPONDER UE doit utiliser les paramètres suivants obtenus à partir de la primitive de service *GetGroupInd*, afin d'identifier les attributs de groupe à renvoyer avec la primitive de service *GetGroupResp* (Règle c) ci-dessous).

Paramètre	Paramètre utilisé pour la recherche dans le CS(R)
Group-type	Group-type
Group-number	Group-number

⁷⁵⁾ Noter que zéro objet dans Objid est légal dans le contexte de la *Group Readout FU*.

⁷⁶⁾ Les Object numbers doivent être assignés aux Object identifiers décodés dans l'ordre numérique croissant habituel, en commençant par la valeur du paramètre Index1 obtenue à partir de la transaction.

⁷⁷⁾ Toutefois, avec les groupes pour lesquels le nombre d'Object identifiers définis est égal à la valeur de l'attribut Group size (groupes ayant atteint leur nombre maximal d'objets), la transaction supplémentaire n'est pas effectuée.

LOOP:

IF current object count is less than value of Group size, as determined from the first transaction:

Perform new transaction, with:

Index1 = current object count, plus one

Index2 = value of Group size, as determined from the first transaction

IF Objid parameter is not empty (contains at least one Object identifier)⁷⁵⁾:

Register, from transaction, component attribute Object identifier for each object, as decoded from the Objid parameter.⁷⁶⁾

Update current object count, adding the new number of objects decoded from the Objid parameter to the previous value of current object count.

ELSE

Exit LOOP

END-IF

ELSE

Exit LOOP

END-IF

END-LOOP

Terminate the FU invocation.

END

In this case, note that the number of transactions required to obtain a complete group readout is normally one larger than the number of transactions required to carry the actual set of Object identifiers: the last transaction will usually contain zero Object identifiers⁷⁷⁾.

This also applies to groups for which the value of attribute Group size is not larger than allowing the *GetGroupConf* PDU to hold the complete configuration, and to groups for which the value of attribute Group size is larger than allowing the *GetGroupConf* PDU to hold the complete configuration but the actual number of Object identifiers being small enough to allow it:

In both cases, the *Group Readout FU* invocation will encompass two *GetGroup* transactions.

The underlying reason is that a *GetGroup* transaction has no way of simultaneously reporting a complete configuration, and the fact that the configuration is complete.

- b) The RESPONDER UE shall use the following parameters from the *GetGroupInd* service primitive, in order to identify the group attributes to be returned with the *GetGroupResp* service primitive (Rule c), below):

Parameter	Attribute used for search in the CS(R)
Group-type	Group type
Group-number	Group number

⁷⁵⁾ Note that zero objects in Objid is legal within the context of the *Group Readout FU*.

⁷⁶⁾ Object numbers shall here be assigned to the decoded Object identifiers in the usual ascending numerical sequence, starting with the value of parameter Index1 from the transaction.

⁷⁷⁾ However, with groups for which the number of defined Object identifiers is equal to the value of attribute Group size (groups having their maximum number of objects), the additional transaction will not be performed.

- c) Le RESPONDER UE doit copier les valeurs des attributs de groupe suivants dans les paramètres suivants de la primitive de service *GetGroupResp*:

Attribut	Paramètre
Group type	Group-type
Group number	Group-number
Persistent	Persist
Static	Static
Priority	Priority
Group size	Max Obj
Object identifier size	Max Obj Size

Le jeu complet des attributs Object identifier de l'attribut Object set, ou un sous-ensemble de ce dernier, doit être codé dans le paramètre Objid(I). Les valeurs croissantes consécutives d'index (I)⁷⁸⁾ dans Objid(I) doivent correspondre à des valeurs croissantes consécutives d'Object number dans le CS(R) pour les mêmes objets. La première valeur d'Object number doit être:

Si les paramètres Index1 = Index2 = 0:

1.

Sinon:

Valeur d'Index1.

- d) Si Index1 >< 0 et Index2 >< 0 et si l'INITIATOR UE demande des Object identifiers pour davantage d'objets⁷⁹⁾ que le nombre actuellement défini dans le CS(R), le RESPONDER doit répondre avec les valeurs d'Index1 et Index2 contenues dans la demande, mais avec le paramètre Objid(I) ne contenant pas plus d'Object identifiers que le nombre actuellement défini dans le CSR, en commençant à la valeur d'Object number dans Index1. Cas particulier, Objid(I) ne doit contenir aucun Object identifiers si Index1 proprement dit est trop grand.

Si les paramètres Index1 et Index2 sont par ailleurs valides, le CS(R) ne contient pas le nombre requis d'objets: le RESPONDER UE ne doit donc jamais renvoyer une valeur différente de *result-ok* dans le paramètre Result.

- e) Le RESPONDER UE doit toujours entrer autant d'Object identifiers que possible dans le paramètre Objid, lorsque Index1 = Index2 = 0.
- f) Le cas décrit à la règle d) ci-dessus doit être le seul cas où le nombre d'Object identifiers renvoyé dans le paramètre Objid(I) ne correspond pas aux valeurs d'Index1 et Index2 alors que le paramètre Result = *result-ok*. Pour d'autres valeurs de Result, voir 6.4.3.2.
- g) Les règles indiquées en 6.1 s'appliquent également ici.

⁷⁸⁾ Cet index n'est pas à interpréter comme un index dans un tableau Objid(I) au sens habituel, mais seulement comme un nombre entier d'ordonnement pour les éléments codés dans le paramètre Objid. Les éléments (Object identifiers) sont des chaînes de texte ASCII, toutes codées dans un tableau d'octets unidimensionnel ; voir A.2.4.1 de la CEI 60870-6-501). Augmenter la valeur de "index I" correspond à passer à une valeur d'index supérieure dans ce tableau d'octets.

⁷⁹⁾ Le nombre d'objets requis est donné par Index2 – Index1 + 1, si Index1 et Index2 ne sont ni l'un ni l'autre égaux à zéro.

- c) The RESPONDER UE shall copy the values of the following group attributes into the following parameters of the *GetGroupResp* service primitive:

Attribute	Parameter
Group type	Group-type
Group number	Group-number
Persistent	Persist
Static	Static
Priority	Priority
Group size	Max Obj
Object identifier size	Max Obj Size

The whole set of component attributes Object identifier within the attribute Object set, or a subset thereof, shall be coded into the Objid(I) parameter. Consecutive ascending index (I)⁷⁸⁾ values in Objid(I) shall correspond to consecutive ascending values of Object number in the CS(R), for the same objects. The first Object number value shall be:

If parameters Index1 = Index2 = 0:

1.

If not:

Value of Index1.

- d) If Index1 >> 0 and Index2 >> 0 and the INITIATOR UE is requesting Object identifiers for more objects⁷⁹⁾ than the number currently defined in the CS(R), the RESPONDER UE shall respond with the values of Index1 and Index2 contained in the request, but with the Objid(I) parameter containing no more Object identifiers than the number currently defined in the CS(R), starting at the Object number value in Index1. As a special case, Objid(I) shall contain no Object identifiers at all, if Index1 itself is too large.

The RESPONDER UE shall never return a value different from *result-ok* in parameter Result because of the CS(R) not containing the requested number of objects, if the Index1 and Index2 parameters are otherwise valid.

- e) The RESPONDER UE shall always pack the maximum possible number of Object identifiers into the Objid parameter, when Index1 = Index2 = 0.
- f) The case described in rule d) above shall be the only case in which the number of Object identifiers returned in the Objid(I) parameter does not match the values of Index1 and Index2, while parameter Result = *result-ok*. For other values of Result, see 6.4.3.2.
- g) The rules stated in 6.1 also apply here.

⁷⁸⁾ This index is not to be interpreted as an index into an array Objid(I) in the usual sense, but only as an integer ordering number for the elements coded into the Objid parameter. The elements (Object identifiers) are ASCII text strings, all consecutively coded into a one-dimensional array of octets; see IEC 60870-6-501, A.2.4.1). Increasing the value of "index I" corresponds to moving to a greater index value within this octets array.

⁷⁹⁾ Requested number of objects is given by Index2 – Index1 + 1, if both Index1 and Index2 are not equal to zero.

6.4.3.1.2 Parameter values

GetGroup:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Group-type	Valeur <i>Measurand-group, Status-group, Discrete-group, Logical-breaker-status-group, Binary-command-group, Analog-setpoint-group, Digital-setpoint-group</i> ou <i>Text-message-group</i> . La valeur doit être égale à la valeur donnée de l'attribut Group type du CS(R) pour le groupe n° Group-number.	Copie de la valeur à partir de <i>Ind</i>
Group-number	Numéro de référence pour le groupe en question. Valeur de l'attribut Group number dans le jeu d'attributs du CS(R), pour laquelle les valeurs de l'attribut sont à renvoyer si possible.	Copie de la valeur à partir de <i>Ind</i>
Persist	(Non applicable)	Valeur de l'attribut Persistent pour le groupe n° Group-number du CS(R)
Static	(Non applicable)	Valeur de l'attribut Static pour le groupe n° Group-number du CS(R)
Priority	(Non applicable)	Valeur de l'attribut Priority pour le groupe n° Group-number du CS(R)
Max Obj	(Non applicable)	Valeur de l'attribut Group size pour le groupe n° Group-number du CS(R)
Index1	Borne inférieure de la plage de valeurs contiguës de l'attribut Object number pour lequel la valeur correspondante de l'attribut Object identifier du CS(R) est à renvoyer par le biais du paramètre <i>Objid(l)</i> . Doit être = 0 si Index2 = 0, ce qui signifie renvoyer autant d'Object identifiers que possible. Si > 0: Doit être >= 1 et <= à la valeur donnée de l'attribut Group size pour le groupe, dans le CS(R).	Copie de la valeur à partir de <i>Ind</i>
Index2	Borne supérieure de la plage de valeurs contiguës de l'attribut Object number pour lequel la valeur correspondante de l'attribut Object identifier du CS(R) est à renvoyer par le biais du paramètre <i>Objid(l)</i> . Doit être = 0 si Index1=0, ce qui signifie renvoyer autant d'Object identifiers que possible. Si > 0: Doit être >= Index1 et <= à la valeur donnée de l'attribut Group size pour le groupe, dans le CS(R).	Copie de la valeur à partir de <i>Ind</i>
Max Obj Size	(Non applicable)	Valeur de l'attribut Object identifier size pour le groupe n° Group-number du CS(R)

6.4.3.1.2 Parameter values

GetGroup:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Group-type	Value <i>Measurand-group, Status-group, Discrete-group, Logical-breaker-status-group, Binary-command-group, Analog-setpoint-group, Digital-setpoint-group</i> or <i>Text-message-group</i> . Value shall be equal to assumed value of attribute Group type in the CS(R) for group No. Group-number.	Copy of value from <i>Ind</i>
Group-number	Reference number for the group in question. Value of attribute Group number in set of attributes in the CS(R), for which attribute values are to be returned, if possible.	Copy of value from <i>Ind</i>
Persist	(Not applicable)	Value of attribute Persistent for group No. Group-number in CS(R)
Static	(Not applicable)	Value of attribute Static for group No. Group-number in CS(R)
Priority	(Not applicable)	Value of attribute Priority for group No. Group-number in CS(R)
Max Obj	(Not applicable)	Value of attribute Group size for group No. Group-number in CS(R)
Index1	Lower bound of contiguous range of values of component attribute Object number for which the corresponding value of component attribute Object identifier in the CS(R) is to be returned via parameter Objid(I). Shall be = 0, if Index2 = 0, signifying <i>return as many Object identifiers as possible</i> . If >< 0: Shall be >= 1 and <= assumed value of attribute Group size for the group, in the CS(R).	Copy of value from <i>Ind</i>
Index2	Upper bound of contiguous range of values of component attribute Object number for which the corresponding value of component attribute Object identifier in the CS(R) is to be returned via parameter Objid (I). Shall be = 0, if Index1 = 0, signifying <i>return as many Object identifiers as possible</i> . If >< 0: Shall be >= Index1 and <= assumed value of attribute Group size for the group, in the CS(R)	Copy of value from <i>Ind</i>
Max Obj Size	(Not applicable)	Value of attribute Object identifier size for group No. Group-number in CS(R)

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Objid	(Non applicable)	Jeu des valeurs des attributs Object identifier du groupe n° Groupnumber du CS(R), ou sous-ensemble de ce dernier. Les valeurs sont à placer aux index croissants consécutifs (I)⁸⁰⁾, par ordre croissant consécutif de la valeur de leur attribut Object number dans le CS(R). Si Index1 = Index2 = 0: Doit contenir autant de valeurs que possible, en commençant par Object number 1. Le facteur limitant doit être la taille maximale du paramètre Objid ou le nombre de valeurs contenues dans le CS(R), pour le groupe. Sinon: doit contenir le nombre de valeurs suivant: (Index2 – Index1) + 1, ou moins, en commençant par Object number = Index1. Si le nombre est inférieur, le facteur limitant doit être la valeur maximale d'Object number dans le CS(R), pour le groupe. Les règles de codage se trouvent en A.2.4.3 de la CEI 60870-6-501. Pour la security class 2: le code d'authentification est ajouté après l'octet zéro de fin. Le code d'authentification est généré sur la base du paramètre Objid(I), en excluant l'octet zéro de fin. La taille de l'Objid(I) plus le code d'authentification ne doit pas dépasser la taille maximale. Pour la security class 3: le paramètre Objid(I) est crypté, à l'exclusion de l'octet zéro de fin. Une somme de contrôle doit être présente. Diverses manières de procéder sont décrites en 9.5.
Result	(Non applicable)	Action effectuée comme spécifié (les paramètres Persist, Static, Priority, Gsize, Objlength et Objid (I) sont valides): <i>result-ok</i>. Action non effectuée, du fait d'une condition d'erreur (les paramètres Persist, Static, Priority, Gsize, Objlength et Objid (I) ne sont pas valides): Code d'erreur: Valeur >< <i>result-ok</i>. Voir 6.4.3.2.

6.4.3.2 Traitement des erreurs

6.4.3.2.1 FU disruption

Interruption par la *Permanent Association FU* ou la *Dynamic Association FU*:

L'interruption de la partie INITIATOR et de la partie RESPONDER de l'invocation en cours de la *Group Readout FU* doit être déclenchée localement, dans le cadre du traitement des primitives *AbortInd* entrantes dans la partie INITIATOR et dans la partie RESPONDER de l'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* qui traite l'association où la *Group Readout FU* est en fonctionnement.

⁸⁰⁾ Voir note 78.

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Objid(I)	(Not applicable)	The set of component attributes Object identifier for group No. Group-number in CS(R), or a subset thereof. The values are to be placed in ascending consecutive indices, 1,⁸⁰⁾ in ascending consecutive order of value of their attribute Object number from the CS(R). If Index1 = Index2 = 0: shall contain as many values as possible, starting with Object number 1. The limiting factor shall be the maximum size of the Objid parameter or the number of values contained in the CS(R), for the group. If not: shall contain the following number of values: (Index2 – Index1) + 1, or fewer, starting with Object number = Index1. If fewer, the limiting factor shall be the maximum value of Object number in the CS(R), for the group. Coding rules are to be found in IEC 60870-6-501, A.2.4.3. For security class 2: the authentication code is appended after the terminating zero octet. The authentication code is generated on the basis of the Objid(I) parameter excluding the terminating zero octet. The size of the Objid(I) plus the authentication code must not exceed the maximum size. For security class 3: The Objid(I) parameter excluding the terminating zero octet is enciphered. A checksum must be present. Possible ways to do this are described in 9.5.
Result	(Not applicable)	Action performed as specified (parameters Persist, Static, Priority, Gsize, Objlength and Objid (I) are valid): <i>result-ok</i> Action not performed, due to error condition (parameters Persist, Static, Priority, Gsize, Objlength and Objid(I) are not valid): Error code: Value >> <i>result-ok</i>. See 6.4.3.2.

6.4.3.2 Error handling

6.4.3.2.1 FU disruption

Disruption by the *Permanent Association FU* or *Dynamic Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Group Readout FU* shall be triggered locally, as a part of the handling of incoming *AbortInd* primitives in both the INITIATOR part and the RESPONDER part of the *Permanent Association FU* or *Dynamic Association FU* invocation handling the association where the *Group Readout FU* is running.

⁸⁰⁾ See footnote 78).

Les deux parties de l'invocation en cours de la *Group Readout FU* doivent être terminées correctement, sans que l'une ou l'autre des parties n'essaie d'émettre une primitive associée à la fin elle-même.⁸¹⁾

6.4.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Group Readout FU* n'est pas présente dans un INITIATOR UE:

Le traitement de ce type d'erreur résultant d'une tentative d'invocation par la Coordinating Function locale constitue un problème local qui ne fait pas partie du domaine d'application du présent rapport technique.

Si ce type d'erreur survient lors d'une tentative d'invocation lors d'une invocation de la *Group Configuration FU*, l'erreur est traitée par le mécanisme de traitement des erreurs de la *Group Configuration FU*. Voir la description de la *Group Configuration FU*.

Si la *Group Readout FU* n'est pas présente dans un RESPONDER UE:

Le RESPONDER UE doit répondre aux tentatives d'invocation de la manière suivante:

- émettre une primitive *GetGroupResp* avec Result = *remote-service-user-unavailable*.

FU présente, mais tentative illégale:

Les tentatives d'invocation sont toujours légales; voir 6.4.2.3.2.

6.4.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>GetGroupConf</i>
FU non en fonctionnement	Ignorer, ou indication d'erreur locale
FU en fonctionnement, en attente de <i>GetGroupConf</i>	(Normal)

Partie RESPONDER:

Non applicable: une primitive de service *GetGroupInd* n'est jamais hors contexte dans le RESPONDER.⁸²⁾

6.4.3.2.4 Timing errors

Erreur dans la partie INITIATOR: Non applicable: aucune contrainte de synchronisation n'est imposée à la partie INITIATOR de la *Group Readout FU*.

⁸¹⁾ Les modes opératoires locaux de nettoyage ne sont pas spécifiés dans le présent rapport technique.

⁸²⁾ L'ordonnancement correct de la primitive *Ind/Resp* est imposé par le TASE.1. Le présent rapport technique ne spécifie aucune imposition de l'ordonnancement des sous-groupes par des transactions *GetGroup* individuelles de la part du RESPONDER UE.

Both parts of the current invocation of the *Group Readout FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself⁸¹⁾.

6.4.3.2.2 Illegal invocation attempt

FU not present:

If the *Group Readout FU* is not present in an INITIATOR UE:

The handling of this type of error resulting from an invocation attempt by the local Coordinating Function is a local issue, outside the scope of this technical report.

If this type of error occurs upon an invocation attempt by a *Group Configuration FU* invocation, the error is handled by the error handling mechanism within the *Group Configuration FU*. See the *Group Configuration FU* description.

If the *Group Readout FU* is not present in a RESPONDER UE:

The RESPONDER UE shall respond to activation attempts in the following way:

- Issuing a *GetGroupResp* primitive with Result = *remote-service-user-unavailable*.

FU present, but attempt illegal:

Invocation attempts are always legal; see 6.4.2.3.2.

6.4.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>GetGroupConf</i>
FU not running	Ignore, or local error indication
FU running, waiting for <i>GetGroupConf</i>	(Normal)

RESPONDER part: Not applicable: a *GetGroupInd* service primitive is never out of context in the RESPONDER⁸²⁾.

6.4.3.2.4 Timing errors

Error in INITIATOR part: Not applicable: There are no timing restraints placed upon the INITIATOR part of the *Group Readout FU*.

⁸¹⁾ Local clean-up procedures are not specified by this technical report.

⁸²⁾ Proper *Ind/Resp* primitive sequencing is enforced by the TASE.1. This technical report does not specify any enforcement of subgroup sequencing across individual *GetGroup* transactions on the part of the RESPONDER UE.

Erreur dans la partie RESPONDER:

Erreur	Réaction de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à <i>GetGroupInd</i>	Sur le RESPONDER: Erreur locale lors d'une éventuelle tentative d'émission de <i>GetGroupResp</i> Sur l'INITIATOR: <i>GetGroupConf</i>, avec Result = <i>remote-service-user-unavailable</i>.	Partie RESPONDER: Terminer l'invocation de la FU localement . Partie INITIATOR: Terminer l'invocation de la FU localement .

6.4.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsqu'elle se produit lors d'une tentative de *GetGroupReq*:

- terminer l'invocation de la FU localement .

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsqu'elle se produit lors d'une tentative de *GetGroupResp*:

- terminer l'invocation de la FU localement .

Voir 4.2.1.

6.4.3.2.6 TASE.1 service primitive parameters errors

Erreurs dans la primitive *GetGroupInd* (détectées par la partie RESPONDER):

La partie RESPONDER doit dans tous les cas de ce type émettre *GetGroupResp* avec Result = <Valeur du tableau ci-dessous>, et les valeurs de Group-number, Group-type, Index1 et Index2 comme dans le *Ind* correspondant.

La partie RESPONDER de la FU ne doit pas être invoquée.

Erreur	Valeur du paramètre Result
Valeur de Gtype non égale à la valeur de l'attribut Group type du CS(R) pour le groupe no. Gnr. (Le cas du groupe n° Gnr non existant est étudié ci-dessous.)	<i>gtype-out-of-range</i>
Valeur de Gnr illégale ou groupe n° Gnr non existant dans le CS(R).	<i>gnr-out-of-range</i>
Index1 = 0 et Index2 >> 0, ou: Index2 = 0 et Index1 >> 0, ou: Index1 < 0 ou Index2 < 0	<i>index-out-of-range</i>
Index1 >> 0 et Index2 < Index1	<i>index-out-of-range</i>
L'Object set du groupe n° Gnr du CS(R) ne peut pas être renvoyé comme spécifié à cause de limitations des ressources locales, ou pour toute raison autre que celles énumérées ci-dessus	<i>remote-service-user-unavailable</i>

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>GetGroupInd</i>	In RESPONDER: Local error from eventual attempt in issuing <i>GetGroupResp</i> In INITIATOR: <i>GetGroupConf</i> , with Result = <i>remote-service-user-unavailable</i> .	<i>RESPONDER part</i> : Terminate FU invocation locally. <i>INITIATOR part</i> : Terminate FU invocation locally.

6.4.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with a *GetGroupReq* attempt:

- terminate FU invocation locally.

See 4.2.1.

RESPONDER part of the FU:

When occurring with a *GetGroupResp* attempt:

- terminate FU invocation locally.

See 4.2.1.

6.4.3.2.6 TASE.1 service primitive parameter errors

Errors in the *GetGroupInd* primitive (detected by the RESPONDER part):

The RESPONDER part shall in all these cases issue *GetGroupResp* with

Result = <Value from table below>, and the values of Group-number, Group-type, Index1 and Index2 as in the corresponding *Ind*

The RESPONDER part of the FU shall not be invoked.

Error	Value of parameter Result
Gtype value not equal to value of attribute Group type in CS(R) for group No. Gnr. (The case of non-existing group No. Gnr is considered below.)	<i>gtype-out-of-range</i>
Value of Gnr is illegal, or group No. Gnr does not exist in the CS(R).	<i>gnr-out-of-range</i>
Index1 = 0 and Index2 >< 0, or: Index2 = 0 and Index1 >< 0, or: Index1 < 0 or Index2 < 0	<i>index-out-of-range</i>
Index1 >< 0 and Index2 < Index1	<i>index-out-of-range</i>
The Object set of group No. Gnr in CS(R) cannot be returned as specified, because of local resource limitations, or any other reason than those listed above	<i>remote-service-user-unavailable</i>

Erreurs dans la primitive *GetGroupConf* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Result = <i>result-ok</i> , et absence de correspondance entre Group-type/Index1/Index2 dans la primitive et Group-type/Index1/Index2 dans la <i>GetGroupReq</i> correspondante	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Result = <i>result-ok</i> , et erreur de format dans Objid	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Result = <i>result-ok</i> , et Index1 >< 0 et Index2 >< 0, et nombre d'Object identifiants codés dans Objid supérieur à la valeur Index2 – Index1 + 1	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Result = <i>result-ok</i> , et nombre d'Object identifiants totalisé pendant l'invocation de la FU en cours supérieur à la valeur de l'attribut Group size pour le groupe n° Group-number du CS(I)	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Result >< <i>result-ok</i>	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Pour la security class 2: le code d'authentification reçu est >< au code d'authentification généré basé sur les données reçues	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Pour la security class 3: la somme de contrôle reçue est >< à la somme de contrôle générée pendant le décryptage	Terminer l'invocation de la FU localement, en enregistrant l'erreur.

6.5 Group Configuration FU

Type: Composite.

6.5.1 Fonction

Une invocation de la *Group Configuration FU* suit la procédure suivante, pour un groupe unique spécifié:

- L'INITIATOR UE invoque la *Group Management FU* pour le groupe, avec la fonction *G-Create*, afin d'établir un descripteur pour le groupe dans le CS(R) du système RESPONDER.
- L'INITIATOR UE invoque la *Group Definition FU* pour le groupe avec le jeu de définition de l'objet spécifié, afin de définir le groupe dans le CS(R).

6.5.2 Règles de coordination

6.5.2.1 Utilisation d'une association

Les interactions TASE.1 faisant partie des invocations *Group Management FU* et *Group Definition FU* invoquées par la *Group Configuration FU* sont véhiculées par des associations possédant les caractéristiques spécifiées pour la *Group Management FU* et la *Group Definition FU*.

6.5.2.2 Relations avec les autres FU

6.5.2.2.1 Invoking FUs

Il est admis d'invoquer la *Group Configuration FU* par la FU suivante:

- *Restart Reconfigure FU*.

Errors in the *GetGroupConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Result = <i>result-ok</i> , and mismatch between Group-type/Index1/Index2 in primitive and Group-type/Index1/Index2 in corresponding <i>GetGroupReq</i>	Terminate the FU invocation locally, registering the error.
Result = <i>result-ok</i> , and format error in Objid	Terminate the FU invocation locally, registering the error.
Result = <i>result-ok</i> , and Index1 >< 0 and Index2 >< 0 and the number of Object identifiers coded into Objid greater than the value Index2 – Index1 + 1	Terminate the FU invocation locally, registering the error.
Result = <i>result-ok</i> , and the number of Object identifiers accumulated during the current FU invocation greater than the value of attribute Group size for group No. Group-number in CS(I)	Terminate the FU invocation locally, registering the error.
Result >< <i>result-ok</i>	Terminate the FU invocation locally, registering the error.
For security class 2: the received authentication code >< the generated authentication code based in the received data	Terminate the FU invocation locally, registering the error.
For security class 3: the received checksum >< the generated checksum during the decipherment	Terminate the FU invocation locally, registering the error.

6.5 Group Configuration FU

Type: Composite.

6.5.1 Function

A *Group Configuration FU* invocation adheres to the following procedure for a single specified group:

- The INITIATOR UE invokes the *Group Management FU* for the group, with function *G-Create*, in order to establish a descriptor for the group in the CS(R) in the RESPONDER system.
- The INITIATOR UE invokes the *Group Definition FU* for the group with the specified object definition set, in order to define the group in the CS(R).

6.5.2 Coordination rules

6.5.2.1 Association usage

The TASE.1 interactions that are part of the *Group Management FU* and *Group Definition FU* invocations invoked by the *Group Configuration FU* are conveyed by associations with the characteristics as specified for the *Group Management FU* and *Group Definition FU*.

6.5.2.2 Relation to other FUs

6.5.2.2.1 Invoking FUs

The *Group Configuration FU* may be invoked by the following FU:

- *Restart Reconfigure FU*

6.5.2.2.2 Invoked FUs

Il est admis que la *Group Configuration FU* invoque les FU suivantes:

- *Group Management FU*;
- *Group Definition FU*.

6.5.2.2.3 Disrupting FUs

Il est admis que la *Group Configuration FU* soit interrompue par l'interruption de l'invocation d'une FU en support.

6.5.2.2.4 Disrupted FUs

La *Group Configuration FU* ne doit interrompre aucune autre FU.

Il est admis toutefois que les FU invoquées par la *Group Configuration FU* interrompent une autre FU quelle qu'elle soit.

6.5.2.3 Invocation

6.5.2.3.1 Conditions préalables

Aucune invocation de FU précédente ou concourante n'est requise⁸³⁾.

6.5.2.3.2 Restrictions

Une invocation de la *Group Configuration FU* en tant que telle est autorisée à tout moment. Elle échouera toutefois si elle viole les règles indiquées en 6.2.2.3.2 et 6.3.2.3.2 relatives aux FU que la *Group Configuration FU* invoque. Les FU invoquées sont données en 6.5.2.2.2.

6.5.2.3.3 Invoking events

Il est autorisé d'invoquer la partie INITIATOR de la *Group Configuration FU* par:

- une demande locale par le biais de la Coordinating Function, dont l'origine est en dehors du domaine d'application du présent rapport technique;
- la *Restart Reconfigure FU*.

La *Group Configuration FU* est de type composite. En conséquence, il n'existe aucune partie RESPONDER spécifique dans la *Group Configuration FU* avec un événement d'invocation associé spécifique.

6.5.2.4 Fin

6.5.2.4.1 Orderly termination

La fin ordonnée de la *Group Configuration FU* doit se produire après la fin ordonnée de la dernière invocation de FU invoquée par la *Group Configuration FU*.

La fin ordonnée inattendue d'une FU quelconque invoquée par la *Group Configuration FU* est considérée comme une erreur dans l'invocation de la *Group Configuration FU*, et doit aussi entraîner la fin ordonnée de la *Group Configuration FU*.

⁸³⁾ Les invocations de la *Permanent Association FU* ou de la *Dynamic Association FU* sont requises par la *Group Management FU* et la *Group Definition FU* invoquées par la *Group Configuration FU*, et non par la *Group Configuration FU* elle-même.

6.5.2.2.2 Invoked FUs

The *Group Configuration FU* may invoke the following FUs:

- *Group Management FU*;
- *Group Definition FU*.

6.5.2.2.3 Disrupting FUs

The *Group Configuration FU* may be disrupted by disruption of a supporting FU invocation.

6.5.2.2.4 Disrupted FUs

The *Group Configuration FU* shall not disrupt any other FU.

The FUs invoked by the *Group Configuration FU* may, however, disrupt any other FU.

6.5.2.3 Invocation

6.5.2.3.1 Prerequisites

No previous or concurrent FU invocations are required⁸³⁾.

6.5.2.3.2 Restrictions

A *Group Configuration FU* invocation as such is allowed at any time. However, it will fail if violating rules stated in the "Restriction", see 6.2.2.3.2 and 6.3.2.3.2 regarding the FUs that the *Group Configuration FU* invokes. Invoked FUs are listed in 6.5.2.2.2.

6.5.2.3.3 Invoking events

The INITIATOR part of the *Group Configuration FU* may be invoked by:

- local request via the Coordinating Function, the original source of which is outside the scope of this technical report;
- the Restart Reconfigure FU.

The *Group Configuration FU* is of the composite type. Consequently, there exists no specific RESPONDER part of the *Group Configuration FU* with an associated specific invoking event.

6.5.2.4 Termination

6.5.2.4.1 Orderly termination

Orderly termination of the *Group Configuration FU* shall take place after orderly termination of the last FU invocation invoked by the *Group Configuration FU*.

Unexpected orderly termination of any FU that the *Group Configuration FU* invokes is considered an error within the *Group Configuration FU* invocation, and shall also lead to orderly termination of the *Group Configuration FU*.

⁸³⁾ *Permanent Association FU* or *Dynamic Association FU* invocations are a requirement of the *Group Management FU* and *Group Definition FU* invoked by the *Group Configuration FU*, not of the *Group Configuration FU* itself.

6.5.2.4.2 Disruption

Il est autorisé d'interrompre la partie INITIATOR d'une invocation de la *Group Configuration FU* par:

- l'interruption d'une autre invocation de FU. Voir 6.5.2.2.3.

La *Group Configuration FU* est de type composite. En conséquence, il n'existe aucune partie RESPONDER spécifique de la *Group Configuration FU* à interrompre.

6.5.3 Procédures

6.5.3.1 Primitives de service TASE.1

La *Group Configuration FU* est de type composite. Aucune séquence de primitive de service TASE.1 spécifique n'y est associée.

6.5.3.1.1 Séquence d'actions

La séquence d'actions normale comprend deux phases:

Phase 1: création du groupe dans le CS(R) du système RESPONDER.

Si les conditions préalables et les restrictions qui s'appliquent à une invocation de la *Group Management FU* pour le groupe peuvent être respectées:

L'INITIATOR UE invoque la *Group Management FU* avec la fonction *G-Create* pour le groupe, avec les valeurs spécifiées des attributs suivants.

- Group type
- Group number
- Group size
- Object identifier size
- Persistent
- Static
- Priority

Si la *Group Management FU* ne fonctionne pas correctement ou si le résultat de la *Group Management FU* n'est pas correct:

L'invocation de la *Group Configuration FU* se termine par une indication d'erreur.

Sinon:

L'invocation de la *Group Configuration FU* se termine par une indication d'erreur.

Phase 2: définition du groupe dans le CS(R) du système RESPONDER.

Si les conditions préalables et les restrictions qui s'appliquent à une invocation de la *Group Definition FU* pour le groupe peuvent être respectées:

L'INITIATOR UE invoque la *Group Definition FU* pour la définition du groupe, avec les valeurs spécifiées des attributs suivants.

- Group type
- Group number
- Object set

Si la *Group Definition FU* ne fonctionne pas correctement ou si le résultat de la *Group Definition FU* n'est pas correct:

L'invocation de la *Group Configuration FU* se termine par une indication d'erreur.

Sinon:

L'invocation de la *Group Configuration FU* se termine par une indication d'erreur.

6.5.2.4.2 Disruption

The INITIATOR part of a *Group Configuration FU* invocation may be disrupted by:

- disruption of another FU invocation. See 6.5.2.2.3.

The *Group Configuration FU* is of the composite type. Consequently, there exists no specific RESPONDER part of the *Group Configuration FU* to be disrupted.

6.5.3 Procedures

6.5.3.1 TASE.1 service primitives

The *Group Configuration FU* is of the composite type, having no specific TASE.1 service primitive sequence associated with it.

6.5.3.1.1 Action sequence

The normal action sequence has two phases:

Phase 1: Creating the group in the CS(R) of the RESPONDER system

If the prerequisites and restrictions that apply to a *Group Management FU* invocation for the group can be met:

The INITIATOR UE invokes the *Group Management FU* with function *G-Create* for the group, with specified values of the following attributes.

- Group type
- Group number
- Group size
- Object identifier size
- Persistent
- Static
- Priority

If the *Group Management FU* is NOT running OK or the *Group Management FU* invocation does NOT complete OK:

The *Group Configuration FU* invocation terminates, with error indication.

If not:

The *Group Configuration FU* invocation terminates, with error indication.

Phase 2: Defining the group in the CS(R) of the RESPONDER system

If the prerequisites and restrictions that apply to a *Group Definition FU* invocation for the group can be met:

The INITIATOR UE invokes the *Group Definition FU* for definition of the group, with specified values of the following attributes.

- Group type
- Group number
- Object set

If the *Group Definition FU* is NOT running OK or the *Group Definition FU* invocation does NOT complete OK:

The *Group Configuration FU* invocation terminates, with error indication.

If not:

The *Group Configuration FU* invocation terminates, with error indication.

6.5.3.1.2 Parameter values

La FU est de type composite. Voir 4.2.2.

6.5.3.2 Traitement des erreurs

Le traitement des erreurs survenant avec les primitives de service TASE.1 n'est pas pris en compte ici, puisque la *Group Configuration FU* est de type composite.

Les erreurs survenant dans les FU invoquées par la *Group Configuration FU* sont décrites en 6.5.3.1.1. Le traitement ultérieur de ces erreurs est d'ordre local dans le INITIATOR UE, et n'est pas spécifié dans le présent rapport technique. Toutefois, les erreurs éventuelles doivent être signalées à la FU effectuant l'invocation.

6.5.3.2.1 FU disruption

L'interruption de la *Group Configuration FU* doit être traitée ultérieurement par l'INITIATOR UE comme pour les autres erreurs de la *Group Configuration FU*; voir 6.5.3.2.

6.5.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Group Configuration FU* n'est pas présente dans un INITIATOR UE:

Les demandes d'invocation sont toujours générées localement; voir 6.5.2.3. En conséquence, le traitement de ce type d'erreur constitue un problème local qui ne fait pas partie du domaine d'application du présent rapport technique. Toutefois, de telles erreurs, si elles se produisent, doivent être signalées à la FU qui effectue l'invocation.

Aucune partie RESPONDER spécifique n'est définie pour la *Group Configuration FU*.

FU présente, mais tentative illégale:

Les tentatives d'invocation de la *Group Configuration FU* en tant que telles sont toujours légales. Voir 6.5.2.3.2.

7 Function group transfert de données

7.1 Requested Data Transfer FU

Type: Primary.

7.1.1 Fonction

Une invocation de la *Requested Data Transfer FU* doit être conforme aux procédures suivantes:

- a) L'INITIATOR UE donne l'instruction au RESPONDER UE de renvoyer un jeu de données spécifié, en respectant les règles générales suivantes:
 - Les données doivent être associées à un seul groupe spécifié, mais doivent contenir le nombre d'incarnations requis. Le groupe entier, ou seulement un sous-ensemble contenant ces données est requis.

6.5.3.1.2 Parameter values

The *Group Configuration FU* is of the composite type. See 4.2.2.

6.5.3.2 Error handling

Handling of errors occurring with TASE.1 service primitives is not considered here, since the *Group Configuration FU* is of the composite type.

Errors occurring in FUs invoked by the *Group Configuration FU* are described in 6.5.3.1.1. Further processing of such errors is a local matter in the INITIATOR UE, not specified by this technical report. However, errors shall be reported to the invoking FU, if any.

6.5.3.2.1 FU disruption

Disruption of the *Group Configuration FU* shall be further processed by the INITIATOR UE as for other *Group Configuration FU* errors; see 6.5.3.2.

6.5.3.2.2 Illegal invocation attempt

FU not present:

If the *Group Configuration FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see 6.5.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this technical report. However, such errors shall be reported to the invoking FU, if any.

No specific RESPONDER part is defined for the *Group Configuration FU*.

FU present, but attempt illegal:

Group Configuration FU invocation attempts as such are always legal. See 6.5.2.3.2.

7 The Data Transfer Function Group

7.1 Requested Data Transfer FU

Type: Primary.

7.1.1 Function

A *Requested Data Transfer FU* invocation shall adhere to the following procedure:

- a) The INITIATOR UE instructs the RESPONDER UE to return a specified set of data, subject to the following general rules:
 - The data shall be associated with one specified group only, but shall comprise the requested number of incarnations. The whole group or a subset thereof may be requested.

- Le même nombre de valeurs de données doit être renvoyé pour chaque invocation.⁸⁴⁾
- La séquence de valeurs des données d'une transmission doit être identique à celle de la définition de groupe.
- Chaque transmission d'un groupe de type *Text-message-group* doit contenir la valeur de données d'un objet et d'un seul. En d'autres termes, pour les groupes de ce type, chaque objet doit constituer un sous-ensemble de données.
- La séquence d'incarnations doit être ordonnée soit selon la date, en commençant par l'incarnation la plus ancienne, soit selon les objets d'un groupe, en commençant par l'index le plus faible. Voir l'annexe A de la CEI 60870-6-502.
- Pour une invocation donnée, le RESPONDER UE doit renvoyer le jeu de valeurs de données qui étaient valides dans le système RESPONDER au moment où elles ont été calculées à partir des paramètres, pour cette invocation.
- Le RESPONDER UE doit toujours renvoyer le nombre d'incarnations spécifié par le paramètre Counts de la requête, sauf lorsque le premier⁸⁵⁾ élément du paramètre T est égal à –1, si une seule incarnation (jeu de valeurs le plus récent) doit être renvoyée.

b) Le RESPONDER UE renvoie les données requises.

7.1.2 Règles de coordination

7.1.2.1 Utilisation d'une association

Toutes les interactions de TASE.1 faisant partie d'une invocation de la *Requested Data Transfer FU* sont véhiculées par une association unique. L'association doit avoir les caractéristiques spécifiées en 7.1.2.3.1.

7.1.2.2 Relation avec les autres FU

7.1.2.2.1 Invoking FUs

Il est permis à la *Periodically Requested Data Transfer FU* d'invoquer la *Requested Data Transfer FU*.

7.1.2.2.2 Invoked FUs

La *Requested Data Transfer FU* ne doit invoquer aucune autre FU.

7.1.2.2.3 Disrupting FUs

Il est permis aux FU suivantes d'interrompre la *Requested Data Transfer FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.1.2.2.4 Disrupted FUs

La *Requested Data Transfer FU* ne doit interrompre aucune autre FU.

⁸⁴⁾ Le protocole ne limite pas la taille des données, mais l'implémentation la limite parfois. Il convient, pour chaque implémentation, d'être au moins capable de répondre en utilisant des données conformes à [1] (voir Bibliographie).

⁸⁵⁾ A la valeur d'index la plus faible.

- The same number of data values shall be returned per invocation⁸⁴⁾.
- The sequence of the data values in a transmission shall be the same as in the group definition.
- For groups of type *Text-message-group*, each transmission shall contain the data value of exactly one object. In other words, for groups of this type each object shall constitute a data subset.
- The sequence of incarnations shall be ordered according to time, oldest incarnation first, or according to objects in the group, lowest index first.
See annex A of IEC 60870-6-502.
- The RESPONDER UE shall, for a given invocation, return the set of data values that were currently valid in the RESPONDER system at the point in time computed from the parameters in the request for that invocation.
- The RESPONDER UE shall always return the number of incarnations specified by the parameter Counts of the request, except when the first⁸⁵⁾ element in parameter T is equal to –1, when only one incarnation (latest set of values) shall be returned.

b) The RESPONDER UE returns the requested data.

7.1.2 Coordination rules

7.1.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Requested Data Transfer FU* are conveyed by one single association. The association shall have the characteristics as specified in 7.1.2.3.1.

7.1.2.2 Relation to other FUs

7.1.2.2.1 Invoking FUs

The *Requested Data Transfer FU* may be invoked by the *Periodically Requested Data Transfer FU*.

7.1.2.2.2 Invoked FUs

The *Requested Data Transfer FU* shall not invoke any other FU.

7.1.2.2.3 Disrupting FUs

The *Requested Data Transfer FU* may be disrupted by:

- *Permanent Association FU*;
- *Dynamic Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.1.2.2.4 Disrupted FUs

The *Requested Data Transfer FU* shall not disrupt any other FU.

⁸⁴⁾ Though the protocol is not limiting the size of data, the implementation may have to. Any implementation should at least be able to respond with data according to [1]. (See Bibliography.)

⁸⁵⁾ At lowest index value.

7.1.2.3 Invocation

7.1.2.3.1 Conditions préalables

Les FU suivantes doivent avoir été invoquées avant toute invocation de la *Requested Data Transfer FU*:

- *Permanent Association FU* ou *Dynamic Association FU*;
- *Group Configuration FU*.⁸⁶⁾

L'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* doit encore être en fonctionnement au moment de l'invocation de la *Requested Data Transfer FU*. Si la *Permanent Association FU* est en fonctionnement, l'association maintenue par l'invocation de la *Permanent Association FU* doit également être en fonctionnement (elle ne doit pas être interrompue temporairement) au moment de l'invocation de la *Requested Data Transfer FU*.

L'invocation de la *Group Configuration FU* doit se terminer lorsque la *Requested Data Transfer FU* est invoquée pour le même groupe.

La *Permanent Association FU* ou la *Dynamic Association FU* doivent avoir été invoquées pour pouvoir créer l'association à utiliser pour les interactions associées à l'invocation en cours de la *Requested Data Transfer FU* (et pour que la *Permanent Association FU* puisse maintenir cette association), avec la paire A-suffix spécifiée dans le tableau de 5.1.

La *Group Configuration FU* doit avoir été invoquée avant de pouvoir créer et définir le groupe à transmettre, et doit se terminer avant l'invocation en cours de la *Requested Data Transfer FU*.⁸⁷⁾

7.1.2.3.2 Restrictions

La *Requested Data Transfer FU* ne doit pas être invoquée tant que l'une au moins des FU suivantes est en cours de fonctionnement, pour le groupe considéré:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

Lorsqu'une *Unsolicited Data Transfer FU* est déjà en cours de fonctionnement dans une association, il est permis à une implémentation de DRFU d'être limitée au seul support des informations présentes (hors informations d'archives)⁸⁸⁾.

7.1.2.3.3 Invoking events

Il est permis aux événements suivants d'invoquer la partie INITIATOR de la *Requested Data Transfer FU*:

- Requête locale par le biais de la Coordinating Function, dont l'origine ne fait pas partie du domaine d'application du présent rapport technique.
- *Periodically Requested Data Transfer FU*.

⁸⁶⁾ Il est également permis à la *Group Management FU* et à la *Group Definition FU*, invoquées par la *Group Configuration FU*, d'être invoquées directement par la Coordinating Function.

⁸⁷⁾ L'utilisation de la *Group Configuration FU* n'est cependant pas nécessaire si le groupe est défini dans le RESPONDER UE par un mécanisme non TASE.1 et également déclaré à l'INITIATOR UE par un mécanisme non TASE.1.

⁸⁸⁾ La *Unsolicited Data* risque autrement d'être retardée par des requêtes de données longues.

7.1.2.3 Invocation

7.1.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Requested Data Transfer FU*:

- The *Permanent Association FU* or *Dynamic Association FU*;
- The *Group Configuration FU* ⁸⁶⁾.

The *Permanent Association FU* or *Dynamic Association FU* invocation shall still be running at the time the *Requested Data Transfer FU* is invoked, and if the *Permanent Association FU* is running, the association maintained by the *Permanent Association FU* invocation shall be running (not temporarily broken) at that time.

The *Group Configuration FU* invocation shall be terminated when the *Requested Data Transfer FU* is invoked for the same group.

The *Permanent Association FU* or *Dynamic Association FU* shall have been invoked to create (and for the *Permanent Association FU*, also to maintain) the association to be used for the interactions related to the current invocation of the *Requested Data Transfer FU*, with the A-suffix pair as specified in the table in 5.1.

The *Group Configuration FU* shall have been invoked to create and define the group that is to be transmitted, and shall be terminated prior to the current invocation of the *Requested Data Transfer FU* ⁸⁷⁾.

7.1.2.3.2 Restrictions

The *Requested Data Transfer FU* shall not be invoked while at least one of the following FUs are running for the group involved:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

On an association where an *Unsolicited Data Transfer FU* is already running, an implementation of the DRFU may be restricted to only support present information (no archive information) ⁸⁸⁾.

7.1.2.3.3 Invoking events

The INITIATOR part of the *Requested Data Transfer FU* may be invoked by:

- Local request via the Coordinating Function, the original source of which is outside the scope of this technical report.
- The *Periodically Requested Data Transfer FU*.

⁸⁶⁾ Alternatively, the *Group Management FU* and *Group Definition FU*, which are invoked by the *Group Configuration FU*, may have been invoked directly by the Coordinating Function.

⁸⁷⁾ The use of the *Group Configuration FU* is, however, not required if the group is defined in the RESPONDER UE by a non-TASE.1 mechanism and also made known to the INITIATOR UE by a non-TASE.1 mechanism.

⁸⁸⁾ Otherwise the Unsolicited Data may be delayed due to long data requests.

L'invocation de la partie RESPONDER de la *Requested Data Transfer FU* est tentée chaque fois qu'une primitive *RequestedDataTransferInd* valide est reçue par le biais d'une association ayant les caractéristiques définies pour la *Requested Data Transfer FU*. Voir 5.5.

7.1.2.4 Fin

7.1.2.4.1 Orderly termination

Il est uniquement permis aux événements suivants de déclencher la fin ordonnée de la partie INITIATOR d'une invocation de la *Requested Data Transfer FU*:

- Requête locale par le biais de la Coordinating Function, dont l'origine ne fait pas partie du domaine d'application du présent rapport technique (fin prématurée).
- Nombre d'erreurs accumulées locales trop important. Voir 7.1.3.2.4.
- Erreur d'encombrement. Voir 7.1.3.2.5.
- Réception d'une primitive de service *RequestedDataTransferConf*.

La partie RESPONDER d'une invocation de la *Requested Data Transfer FU* se termine toujours (fin normale) après l'émission d'une primitive *RequestedDataTransferResp*.

7.1.2.4.2 Disruption

Il est permis aux événements suivants d'interrompre les parties INITIATOR et RESPONDER d'une invocation de la *Requested Data Transfer FU*:

- Interruption d'une autre invocation de FU. Voir 7.1.2.2.3.
- Condition d'erreur fatale en fonctionnement. Voir 7.1.3.2.

7.1.3 Procédures

7.1.3.1 Primitives de service TASE.1

Le service TASE.1 élémentaire suivant est utilisé par la *Requested Data Transfer FU*:

- *RequestedDataTransfer*.

7.1.3.1.1 Sequence

La séquence normale de primitives se présente comme suit:

INITIATOR UE	TASE.1	RESPONDER UE
<i>RequestedDataTransferReq</i>	----->	<i>RequestedDataTransferInd</i>
<i>RequestedDataTransferConf</i>	<-----	<i>RequestedDataTransferResp</i>

Les règles définies en 7.1.1 s'appliquent également ici.

Invocation of the RESPONDER part of the *Requested Data Transfer FU* is attempted whenever a valid *RequestedDataTransferInd* primitive is received via an association with the characteristics as defined for the *Requested Data Transfer FU*: see 5.5.

7.1.2.4 Termination

7.1.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of a *Requested Data Transfer FU* invocation may only be triggered by:

- Local request via the Coordinating Function, the original source of which is outside the scope of this technical report (premature termination).
- Local accumulated error count becoming too great. See 7.1.3.2.4.
- Congestion error. See 7.1.3.2.5.
- Reception of a *RequestedDataTransferConf* service primitive.

The RESPONDER part of a *Requested Data Transfer FU* invocation always terminates itself (normal termination) after issuing a *RequestedDataTransferResp*.

7.1.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Requested Data Transfer FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 7.1.2.2.3.
- A fatal error condition encountered during operation. See 7.1.3.2.

7.1.3 Procedures

7.1.3.1 TASE.1 service primitives

The following elementary TASE.1 services are used by the *Requested Data Transfer FU*:

- *RequestedDataTransfer*.

7.1.3.1.1 Sequence

The normal sequence of primitives is as follows:

INITIATOR UE	TASE.1	RESPONDER UE
<i>RequestedDataTransferReq</i>	----->	<i>RequestedDataTransferInd</i>
<i>RequestedDataTransferConf</i>	<-----	<i>RequestedDataTransferResp</i>

The rules stated in 7.1.1 also apply here.

7.1.3.1.2 Parameter values

Requested Data Transfer request:

Paramètre	Req (INITIATOR)
Gtype	Valeur <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> ou <i>Text-message-group</i> , selon le type du groupe considéré, tel qu'il est défini dans le CS(R).
Gnr	Numéro de référence du groupe considéré, tel qu'il est défini dans le CS(R).
Index1	Borne inférieure de la plage de valeurs adjacente de l'attribut Object numbers pour lequel les données sont à renvoyer, pour chaque incarnation demandée. Doit être = 0 si Index2 = 0, qui signifie <i>renvoyer les données pour tous les Object identifiants définis du groupe</i> . Si >< 0: Doit être >= 1 et <= valeur donnée de l'attribut Group size du groupe, dans le CS(R).
Index2	Borne supérieure de la plage de valeurs adjacente de l'attribut Object numbers pour lequel les données sont à renvoyer, pour chaque incarnation demandée. Doit être = 0, si Index1 = 0, qui signifie <i>renvoyer les données pour tous les Object identifiants définis du groupe</i> . Si >< 0: Doit être >= Index1 et <= valeur supposée de l'attribut Group size du groupe, dans le CS(R).
T	Date et heure de l'incarnation de données demandée la plus ancienne, codée sous forme d'un tableau d'entiers conformément à la CEI 60870-6-501. Il est conseillé d'utiliser l'UTC. Le premier élément du tableau (à l'index le plus faible) égal à -1 doit correspondre uniquement à la dernière requête d'incarnation.
DeltaT	Intervalle de temps requis entre des incarnations de données successives. Peut prendre toute valeur acceptée par la TAPI si le premier élément de T = -1.
T-Unit	Unité utilisée pour le paramètre DeltaT, codée conformément à la CEI 60870-6-501. Peut prendre toute valeur acceptée par la TAPI, si le premier élément de T = -1.
Count	Nombre d'incarnations de données demandées. Peut prendre toute valeur acceptée par la TAPI, si le premier élément de T = -1.

Requested Data Transfer response:

Paramètre	Resp (INITIATOR)
T	Date et heure s'appliquant au jeu complet de valeurs contenues dans le paramètre Data (voir ci-dessous) et déterminées par le RESPONDER UE. Il est conseillé d'utiliser l'UTC.
Dataorg	Spécification indiquant la manière dont les données sont triées si un groupe contenant plusieurs incarnations du jeu de données est transféré; – par heure ou – par objet.
Data	a) Le RESPONDER UE doit rapporter le nombre d'incarnations de données spécifié par le paramètre Count à partir de la requête. Chaque incarnation doit contenir les valeurs de données qui étaient (seront) valides dans le système RESPONDER au moment défini par les paramètres T, DeltaT, T-Unit et Count, à partir de la requête. b) Si au moins l'une des incarnations requises existe dans le système RESPONDER, le RESPONDER UE doit renvoyer le nombre d'incarnations demandé, en fixant le code de qualité à <i>not-ok</i> pour toutes les valeurs de données de toutes les incarnations qui n'existent pas. L'absence d'incarnation est considérée comme une erreur (voir ci-dessous). c) L'absence d'égalité entre l'intervalle de temps spécifié par les paramètres DeltaT et T-Unit et l'intervalle de temps d'archives d'un système RESPONDER est considérée comme une erreur. Voir ci-dessous (pas d'interpolation). La structure est définie dans l'annexe A de la CEI 60870-6-501.
Result	= <i>result-ok</i>

7.1.3.1.2 Parameter values

Requested Data Transfer Request

Parameter	Req (INITIATOR)
Gtype	Value <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> , or <i>Text-message-group</i> , depending on the type of group in question, as assumed in the CS(R).
Gnr	Reference number for the group in question, as assumed in the CS(R).
Index1	Lower bound of adjacent range of values of component attribute Object numbers for which data are to be returned, for each requested incarnation. Shall be = 0, if Index2 = 0, signifying <i>return data for all defined Object identifiers within the group</i> . If >< 0: Shall be >= 1 and <= assumed value of attribute Group size for the group, in the CS(R).
Index2	Upper bound of adjacent range of values of component attribute Object numbers for which data are to be returned, for each requested incarnation. Shall be = 0, if Index1 = 0, signifying <i>return data for all defined Object identifiers within the group</i> . If >< 0: Shall be >= Index1 and <= assumed value of attribute Group size for the group, in the CS(R).
T	Time stamp for oldest requested data incarnation, coded into an integer array according to IEC 60870-6-501. The use of UTC is recommended. First element in array (at lowest index) equal to –1 shall correspond to requesting latest incarnation only.
DeltaT	Requested time interval between successive data incarnations. Can have any value which the TAPI accepts, if first element in T = –1.
T-Unit	Unit for parameter DeltaT, coded according to IEC 60870-6-501. Can have any value which the TAPI accepts, if first element in T = –1.
Count	Requested number of data incarnations. Can have any value which the TAPI accepts, if first element in T = –1.

Requested Data Transfer Response

Parameter	Resp (RESPONDER)
T	Time stamp, applying to the whole set of values contained in the Data parameter (below), and determined by the RESPONDER UE. The use of UTC is recommended.
Dataorg	Specification of how the data is sorted if a group with more than one incarnation of the data set is transferred; – by time or – by object.
Data	a) The RESPONDER UE shall report the number of data incarnations specified by the parameter Count from the request. Each incarnation shall contain the data values that were (will be) valid in the RESPONDER system at the points in time defined by the parameters T, DeltaT, T-Unit and Count from the request. b) If at least one of the requested incarnations does exist in the RESPONDER system, the RESPONDER UE shall return the requested number of incarnations, setting the quality code = <i>not-ok</i> for all data values in all incarnations that do not exist. The case of no existing incarnation is considered an error; see below. c) The case of parameters DeltaT and T-Unit in the request specifying a time interval unequal to the archiving time interval in the RESPONDER system is considered an error; see below (no interpolation). The structure is defined in annex A of IEC 60870-6-501.
Result	= <i>result-ok</i>

7.1.3.2 Traitement des erreurs

7.1.3.2.1 FU disruption

Interruption par la *Permanent Association FU* ou la *Dynamic Association FU*:

L'interruption de la partie INITIATOR et de la partie RESPONDER de l'invocation en cours de la *Requested Data Transfer FU* doit être déclenchée localement, et doit être traitée en même temps que les primitives *TASEAbortInd* entrantes dans la partie INITIATOR et dans la partie RESPONDER de l'invocation de FU traitant l'association sur laquelle la *Requested Data Transfer FU* fonctionne.

Les deux parties de l'invocation en cours de la *Requested Data Transfer FU* doivent se terminer correctement, aucune des deux parties ne devant tenter d'émettre une primitive associée à la fin de l'invocation elle-même.⁸⁹⁾

7.1.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Requested Data Transfer FU* n'est pas présente dans l'INITIATOR UE:

Les requêtes d'invocation sont toujours générées localement, voir 7.1.2.3. En conséquence, ce type d'erreur est traité localement et ne fait pas partie du domaine d'application du présent rapport technique.

Si la *Requested Data Transfer FU* n'est pas présente dans le RESPONDER UE:

Le RESPONDER User Element doit répondre aux tentatives d'activation comme suit:

- émission de la primitive *RequestedDataTransferResp* avec *Result = remote-service-user-unavailable*.

FU présente, mais tentative illégale:

Les tentatives d'invocation illégales de la *Requested Data Transfer FU* pour un groupe quel qu'il soit d'un INITIATOR UE sont traitées au niveau local et ne font pas partie du domaine d'application du présent rapport technique.

Les tentatives d'invocation illégales de la *Requested Data Transfer FU* pour un groupe quel qu'il soit d'un RESPONDER UE doivent être traitées par le RESPONDER UE comme suit:

- émission d'une primitive *RequestedDataTransferResp* avec *Result = remote-service-user-unavailable*, sans (ré)invocation de la *Requested Data Transfer FU* pour le groupe considéré, dans le RESPONDER UE.

7.1.3.2.3 Incoming TASE.1 service primitive out of context

Aucune.

7.1.3.2.4 Timing errors

Erreur dans la partie INITIATOR:

Aucune.

⁸⁹⁾ Les procédures de nettoyage locales ne sont pas spécifiées dans le présent rapport technique.

7.1.3.2 Error handling

7.1.3.2.1 FU disruption

Disruption by the *Permanent Association FU* or the *Dynamic Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Requested Data Transfer FU* shall be triggered locally, as a part of the handling of incoming *TASEAbortInd* primitives in both the INITIATOR part and the RESPONDER part of the FU invocation handling the association on which the *Requested Data Transfer FU* is running.

Both parts of the current invocation of the *Requested Data Transfer FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself⁸⁹⁾.

7.1.3.2.2 Illegal invocation attempt

FU not present:

If the *Requested Data Transfer FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see 7.1.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this technical report.

If the *Requested Data Transfer FU* is not present in RESPONDER UE:

The RESPONDER User Element shall respond to activation attempts in the following way:

- Issuing *RequestedDataTransferResp* primitive with Result = *remote-service-user-unavailable*

FU present, but attempt illegal:

Attempts of illegal invocations of the *Requested Data Transfer FU* for any given group in an INITIATOR UE is a local issue, outside the scope of this technical report.

Attempts of illegal invocations of the *Requested Data Transfer FU* for any given group in a RESPONDER UE shall be handled by the RESPONDER UE in the following way:

- Issuing *RequestedDataTransferResp* primitive with Result = *remote-service-user-unavailable*, without (re-)invoking the *Requested Data Transfer FU* for the group concerned, in the RESPONDER UE.

7.1.3.2.3 Incoming TASE.1 service primitive out of context

None.

7.1.3.2.4 Timing errors

Error in INITIATOR part:

None.

⁸⁹⁾ Local clean-up procedures are not specified by this technical report.

Erreur dans la partie RESPONDER:

Erreur	Réaction de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à la primitive <i>RequestedDataTransferInd</i>	Sur le RESPONDER: Erreur locale lors d'une éventuelle tentative d'émission de <i>RequestedDataTransferResp</i> Sur l'INITIATOR: <i>RequestedDataTransferConf</i> avec Result = <i>remote-service-user-unavailable</i>	Partie RESPONDER: Terminer l'invocation de la FU localement. Partie INITIATOR: Terminer l'invocation de la FU localement.

7.1.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsque ce type d'erreur intervient lors d'une tentative d'émission de *RequestedDataTransferReq*:

- terminer l'invocation de la FU localement.

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsque ce type d'erreur intervient lors d'une tentative d'émission de *RequestedDataTransferResp*:

- terminer l'invocation de la FU localement. (La partie INITIATOR de la FU sera éventuellement terminée par les réactions intervenant après l'expiration du délai du fournisseur de TASE.1.)

Voir 4.2.1.

7.1.3.2.6 TASE.1 service primitive parameter errors

Erreurs dans la primitive *RequestedDataTransferInd* (détectées par la partie RESPONDER):

Dans tous ces cas, la partie RESPONDER doit émettre une primitive *RequestedDataTransferResp* avec Result = <Valeur extraite du tableau ci-dessous>.

La partie RESPONDER de la FU ne doit pas être en fonctionnement.

Erreur	Valeur du paramètre Result dans la <i>RequestedDataTransferResp</i>
Valeur Gtype non égale à la valeur de l'attribut Group type dans CS(R) pour le groupe n° Gnr. (le cas de la non-existence du groupe n° Gnr est étudié ci-dessous.)	<i>gtype-out-of-range</i>
La valeur de Gnr est illégale ou le groupe n° Gnr n'existe pas dans le CS(R).	<i>gnr-out-of-range</i>
Index1 = 0 et Index2 >> 0, ou Index2 = 0 et Index1 >> 0, ou Index1 < 0 ou Index2 < 0	<i>index-out-of-range</i>
Index1 >> 0 et Index2 < Index1	<i>index-out-of-range</i>
Index2 > nombre d'objets configurés	<i>index-out-of-range</i>

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>RequestedDataTransferInd</i>	In RESPONDER: Local error from eventual attempt to issue <i>RequestedDataTransferResp</i> In INITIATOR : <i>RequestedDataTransferConf</i> with Result = <i>remote-service-user-unavailable</i>	RESPONDER part: Terminate FU invocation locally. INITIATOR part: Terminate FU invocation locally.

7.1.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with a *RequestedDataTransferReq* attempt:

- Terminate FU invocation locally.

See 4.2.1.

RESPONDER part of the FU:

When occurring with a *RequestedDataTransferResp* attempt:

- Terminate FU invocation locally. (Initiator part of FU will eventually be terminated upon reactions after timeout in the TASE.1 provider.)

See 4.2.1.

7.1.3.2.6 TASE.1 service primitive parameter errors

Errors in the *RequestedDataTransferInd* primitive (detected by the RESPONDER part):

The RESPONDER part shall in all these cases issue one *RequestedDataTransferResp* primitive with Result = <Value from table below>.

The RESPONDER part of the FU shall not be running.

Error	Value of parameter Result in <i>RequestedDataTransferResp</i>
Gtype value not equal to value of attribute Group type in CS(R) for group No. Gnr. (The case of non-existing group No. Gnr is considered below.)	<i>gtype-out-of-range</i>
Value of Gnr is illegal, or group No. Gnr does not exist in the CS(R).	<i>gnr-out-of-range</i>
Index1 = 0 and Index2 >< 0, or: Index2 = 0 and Index1 >< 0, or: Index1 < 0 or Index2 < 0	<i>index-out-of-range</i>
Index1 >< 0 and Index2 < Index1	<i>index-out-of-range</i>
Index2 > number of objects configured	<i>index-out-of-range</i>

Erreur	Valeur du paramètre Result dans la <i>Requested DataTransferResp</i>
Erreur de format dans T	<i>T-out-of-range</i>
Aucune incarnation de données demandée n'existe dans le système RESPONDER	<i>T-out-of-range</i>
DeltaT et T-Unit spécifient une période non égale à la période d'archives du système RESPONDER	<i>DeltaT-out-of-range</i>
Les données ne peuvent pas être renvoyées comme spécifié en raison de limites de ressources locales ou toute raison autre que celles énumérées ci-dessus	<i>remote-service-user-unavailable</i>

Erreurs dans la primitive *RequestedDataTransferConf* (détectées par la partie INITIATOR):

La partie INITIATOR doit dans tous les cas:

- ignorer les données mais continuer normalement, en incrémentant (facultatif) le compteur d'erreurs et/ou rapporter ou consigner l'erreur dans le journal.

7.2 Periodically Requested Data Transfer FU

Type: Composite.

7.2.1 Fonction

Une invocation de la *Periodically Requested Data Transfer FU* est conforme à la procédure suivante, pour un groupe spécifié unique:

- L'INITIATOR UE invoque la *Requested Data Transfer FU* pour le groupe, afin d'extraire du système RESPONDER le jeu d'incarnations de données spécifié.
- L'INITIATOR UE attend pendant un moment déterminé localement, puis reprend à l'étape a).

Les répétitions se poursuivent jusqu'à ce que l'invocation de la *Periodically Requested Data Transfer FU* soit terminée.

7.2.2 Règles de coordination

7.2.2.1 Utilisation d'une association

Les interactions de TASE.1 qui font partie des invocations de la *Requested Data Transfer FU* invoquées par la *Periodically Requested Data Transfer FU* sont véhiculées par des associations dont les caractéristiques sont spécifiées pour la *Requested Data Transfer FU*.

7.2.2.2 Relation avec les autres FU

7.2.2.2.1 Invoking FUs

La *Periodically Requested Data Transfer FU* ne doit être invoquée par aucune autre FU.

7.2.2.2.2 Invoked FUs

La *Periodically Requested Data Transfer FU* doit invoquer la *Requested Data Transfer FU*.

7.2.2.2.3 Disrupting FUs

Il est permis à une interruption d'invocation de FU d'interrompre la *Periodically Requested Data Transfer FU*.

Error	Value of parameter Result in RequestedDataTransferResp
Format error in T	<i>T-out-of-range</i>
None of the requested data incarnations exist in the RESPONDER system	<i>T-out-of-range</i>
DeltaT and T-Unit specify a period unequal to the archive period in the RESPONDER system	<i>DeltaT-out-of-range</i>
Data cannot be returned as specified, because of local resource limitations, or any other reason than those listed above	<i>remote-service-user-unavailable</i>

Errors in the *RequestedDataTransferConf* primitive (detected by the INITIATOR part):

The INITIATOR part shall in all cases :

- Ignore the data, but otherwise proceed as normal, optionally incrementing the local error count and/or report or log the error.

7.2 Periodically Requested Data Transfer FU

Type: Composite.

7.2.1 Function

A *Periodically Requested Data Transfer FU* invocation complies with the following procedure, for a single specified group:

- The INITIATOR UE invokes the *Requested Data Transfer FU* for the group, in order to fetch the specified set of data incarnations from the RESPONDER system once.
- The INITIATOR UE waits for a locally determined amount of time, and then repeats step a).

The repetitions continue until the *Periodically Requested Data Transfer FU* invocation is terminated.

7.2.2 Coordination rules

7.2.2.1 Association usage

The TASE.1 interactions that are part of the *Requested Data Transfer FU* invocations invoked by the *Periodically Requested Data Transfer FU* are conveyed by associations with the characteristics as specified for the *Requested Data Transfer FU*.

7.2.2.2 Relation to other FUs

7.2.2.2.1 Invoking FUs

The *Periodically Requested Data Transfer FU* shall not be invoked by another FU.

7.2.2.2.2 Invoked FUs

The *Periodically Requested Data Transfer FU* shall invoke the *Requested Data Transfer FU*.

7.2.2.2.3 Disrupting FUs

The *Periodically Requested Data Transfer FU* may be disrupted by disruption of a supporting FU invocation.

7.2.2.2.4 Disrupted FUs

La *Periodically Requested Data Transfer FU* ne doit interrompre aucune autre FU.

Il est cependant permis aux FU invoquées par la *Periodically Requested Data Transfer FU* d'interrompre n'importe quelle autre FU.

7.2.2.3 Invocation

7.2.2.3.1 Conditions préalables

Aucune invocation de FU précédente ou concourante n'est requise⁹⁰⁾.

7.2.2.3.2 Restrictions

Une invocation de la *Periodically Requested Data Transfer FU* en tant que telle est autorisée à tout moment. Elle échouera cependant si elle viole les règles établies dans la section "Restrictions" des descriptions de la FU invoquée par la *Periodically Requested Data Transfer FU*. Voir 7.2.2.2.2.

7.2.2.3.3 Invoking events

Il est permis à l'événement suivant d'invoquer la partie INITIATOR de la *Periodically Requested Data Transfer FU*:

- requête locale, par le biais de la Coordinating Function, dont l'origine ne fait pas partie du domaine d'application du présent rapport technique.

La *Periodically Requested Data Transfer FU* est de type composite. En conséquence, il n'existe pas de partie RESPONDER spécifique de la *Periodically Requested Data Transfer FU* avec un événement d'invocation spécifique associé.

7.2.2.4 Fin

7.2.2.4.1 Orderly termination

La fin ordonnée de la *Periodically Requested Data Transfer FU* doit intervenir après la fin ordonnée de la dernière invocation de FU par la *Periodically Requested Data Transfer FU*, à un moment déterminé par une requête de fin émise par la Coordinating Function locale.

La fin ordonnée inattendue de toute FU invoquée par la *Periodically Requested Data Transfer FU* est considérée comme une erreur dans le cadre de l'invocation de la *Periodically Requested Data Transfer FU*. Il se peut qu'elle conduise ou non à une fin ordonnée de la *Periodically Requested Data Transfer FU*, sur décision locale de l'INITIATOR UE.

7.2.2.4.2 Disruption

Il est permis à l'événement suivant d'interrompre la partie INITIATOR d'une invocation de la *Periodically Requested Data Transfer FU*:

- interruption d'une autre invocation de FU. Voir 7.2.2.2.3.

La *Periodically Requested Data Transfer FU* est de type composite. Il n'existe donc pas de partie RESPONDER spécifique à interrompre dans la *Periodically Requested Data Transfer FU*.

⁹⁰⁾ Les invocations de la *Permanent Association FU* ou de la *Dynamic Association FU* sont requises par la *Requested Data Transfer FU* invoquée par la *Periodically Requested Data Transfer FU*, et non par la *Periodically Requested Data Transfer FU* elle-même.

7.2.2.2.4 Disrupted FUs

The *Periodically Requested Data Transfer FU* shall not disrupt any other FU.

The FUs invoked by the *Periodically Requested Data Transfer FU* may, however, disrupt any other FU.

7.2.2.3 Invocation

7.2.2.3.1 Prerequisites

No previous or concurrent FU invocations are required⁹⁰⁾.

7.2.2.3.2 Restrictions

A *Periodically Requested Data Transfer FU* invocation as such is allowed at any time. However, it will fail if violating rules stated in the “Restrictions” section of the descriptions of the FU that the *Periodically Requested Data Transfer FU* invokes. See 7.2.2.2.

7.2.2.3.3 Invoking events

The INITIATOR part of the *Periodically Requested Data Transfer FU* may be invoked by:

- Local request via the Coordinating Function, the original source of which is outside the scope of this technical report.

The *Periodically Requested Data Transfer FU* is of the composite type. Consequently, there exists no specific RESPONDER part of the *Periodically Requested Data Transfer FU*, with an associated specific invoking event.

7.2.2.4 Termination

7.2.2.4.1 Orderly termination

Orderly termination of the *Periodically Requested Data Transfer FU* shall take place after orderly termination of the last FU invocation invoked by the *Periodically Requested Data Transfer FU*, at a point in time determined by a termination request from the local Coordinating Function.

Unexpected orderly termination of any FU that the *Periodically Requested Data Transfer FU* invokes is considered an error within the *Periodically Requested Data Transfer FU* invocation. It may or may not lead to orderly termination for the *Periodically Requested Data Transfer FU*, subject to local decision within the INITIATOR UE.

7.2.2.4.2 Disruption

The INITIATOR part of a *Periodically Requested Data Transfer FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 7.2.2.2.3.

The *Periodically Requested Data Transfer FU* is of the composite type. Consequently, there exists no specific RESPONDER part of the *Periodically Requested Data Transfer FU* to be disrupted.

⁹⁰⁾ *Permanent Association FU* or *Dynamic Association FU* invocations are a requirement of the *Requested Data Transfer FU* invoked by the *Periodically Requested Data Transfer FU*, not of the *Periodically Requested Data Transfer FU* itself.

7.2.3 Procédures

7.2.3.1 Primitives de service TASE.1

La *Periodically Requested Data Transfer FU* est de type composite et ne dispose pas de séquence de primitives de service TASE.1 associée.

7.2.3.1.1 Séquence d'actions

La séquence d'actions normale se déroule en deux phases qui se répètent jusqu'à ce que la FU soit terminée par l'INITIATOR UE:

Phase 1: Extraction de données spécifiques du système RESPONDER.

Si les conditions préalables et limites applicables à une invocation de la *Requested Data Transfer FU* du groupe sont respectées:

L'INITIATOR UE invoque la *Requested Data Transfer FU* du groupe, en utilisant les paramètres spécifiés définissant les incarnations de données souhaitées pour chaque invocation de la *Requested Data Transfer FU*.

Si la *Requested Data Transfer FU* ne fonctionne PAS correctement ou si l'invocation de la *Requested Data Transfer FU* ne se termine pas correctement:

L'invocation de la *Periodically Requested Data Transfer FU* se termine, avec une indication d'erreur si l'INITIATOR UE l'estime approprié.⁹¹⁾

Sinon:

L'invocation de la *Periodically Requested Data Transfer FU* se termine, avec une indication d'erreur si l'INITIATOR UE l'estime approprié.

Phase 2: Pause.

La longueur de la pause est entièrement déterminée par l'INITIATOR UE. Une fois la pause terminée, la séquence redémarre à la phase 1.

7.2.3.1.2 Parameter values

La FU est de type composite. Voir description des FU.

7.2.3.2 Traitement des erreurs

Le traitement des erreurs intervenant avec les primitives de service TASE.1 n'est pas traité ici, car la *Periodically Requested Data Transfer FU* est de type composite.

Les erreurs intervenant dans les FU invoquées par la *Periodically Requested Data Transfer FU* sont décrites en 7.2.3.1.1. Ces erreurs étant traitées localement dans l'INITIATOR UE; ce traitement n'est pas expliqué dans ce rapport.

7.2.3.2.1 FU disruption

L'interruption de la *Periodically Requested Data Transfer FU* doit ensuite être traitée par l'INITIATOR UE, de même que les autres erreurs de *Periodically Requested Data Transfer FU*. Voir 7.2.3.2.

⁹¹⁾ Par exemple, en raison d'un décompte d'erreurs accumulées trop important dans l'invocation de FU.

7.2.3 Procedures

7.2.3.1 TASE.1 service primitives

The *Periodically Requested Data Transfer FU* is of the composite type, having no specific TASE.1 service primitive sequence associated with it.

7.2.3.1.1 Action sequence

The normal action sequence has two phases that are repeated until the FU is terminated by the INITIATOR UE:

Phase 1: Fetching the specified data from RESPONDER system

If the prerequisites and restrictions that apply to a *Requested Data Transfer FU* invocation for the group can be met:

The INITIATOR UE invokes the *Requested Data Transfer FU* for the group, with specified parameters defining the desired data incarnations per *Requested Data Transfer FU* invocation.

If the *Requested Data Transfer FU* is NOT running OK or the *Requested Data Transfer FU* invocation does NOT complete OK:

The *Periodically Requested Data Transfer FU* invocation terminates, with error indication, if the INITIATOR UE finds this appropriate⁹¹⁾.

If not:

The *Periodically Requested Data Transfer FU* invocation terminates, with error indication, if the INITIATOR UE finds this appropriate.

Phase 2: Pause

The pause length is entirely determined by the INITIATOR UE. After pause expiry, the sequence is restarted at Phase 1.

7.2.3.1.2 Parameter values

The FU is of the composite type. See description of component FUs.

7.2.3.2 Error handling

Handling of errors occurring with TASE.1 service primitives are not considered here, since the *Periodically Requested Data Transfer FU* is of the composite type.

Errors occurring in FUs invoked by the *Periodically Requested Data Transfer FU* are described in 7.2.3.1.1. Further processing of such errors is a local matter in the INITIATOR UE, not specified by this report.

7.2.3.2.1 FU disruption

Disruption of the *Periodically Requested Data Transfer FU* shall be further processed by the INITIATOR UE as for other *Periodically Requested Data Transfer FU* errors; see 7.2.3.2.

⁹¹⁾ For example, because of an accumulated error count within the FU invocation becoming too great.

7.2.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Periodically Requested Data Transfer FU* n'est pas présente dans un INITIATOR UE:

Les requêtes d'invocation sont toujours générées localement. Voir 7.2.2.3. En conséquence, ce type d'erreur est traité localement et ne fait pas partie du domaine d'application du présent rapport technique.

Aucune partie RESPONDER spécifique n'est définie pour la *Periodically Requested Data Transfer FU*, car elle est de type composite.

FU présente, mais tentative illégale:

Les tentatives d'invocation de la *Periodically Requested Data Transfer FU* en tant que telles sont toujours légales. Voir 7.2.2.3.2.

7.3 Unsolicited Data Transfer FU

Type: Primary.

7.3.1 Fonction

Une invocation de la *Unsolicited Data Transfer FU* est conforme aux procédures suivantes:

- a) L'INITIATOR UE autorise le RESPONDER UE à transmettre des données non sollicitées, à condition que les règles générales suivantes soient respectées:
 - L'INITIATOR UE indique au RESPONDER UE laquelle des FU suivantes utiliser pour transmettre des données:
 - La *Unsolicited Periodic Data Transfer FU* ou
 - La *Unsolicited Spontaneous Data Transfer FU*.
 - Les données souhaitées sont associées à un groupe spécifié unique.
 - La séquence de valeurs de données souhaitées est identique à celle de la définition de groupe.
 - Il est permis que la séquence de valeurs de données souhaitées soit un sous-ensemble de la séquence complète. Il est admis que le sous-ensemble varie d'une transmission à l'autre.
 - Il est permis au RESPONDER UE d'utiliser le service de données confirmées, non confirmées ou mixtes spécifié, sur la base d'un groupe. L'INITIATOR UE recommande le service à utiliser.
- b) Il est permis à l'INITIATOR UE de retirer à tout moment l'autorisation d'émettre, en forçant le RESPONDER UE à interrompre la transmission.

7.3.2 Règles de coordination

7.3.2.1 Utilisation d'une association

Toutes les interactions de TASE.1 qui font partie d'une invocation de la *Unsolicited Data Transfer FU* sont véhiculées par une association unique. L'association doit être permanente et doit avoir les caractéristiques spécifiées en 7.3.2.3.1.

7.2.3.2.2 Illegal invocation attempt

FU not present:

If the *Periodically Requested Data Transfer FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see 7.2.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this technical report.

No specific RESPONDER part is defined for the *Periodically Requested Data Transfer FU* as it is of the composite type.

FU present, but attempt illegal:

Periodically Requested Data Transfer FU invocation attempts as such are always legal. See 7.2.2.3.2.

7.3 Unsolicited Data Transfer FU

Type: Primary.

7.3.1 Function

An *Unsolicited Data Transfer FU* invocation complies with the following procedure:

- a) The INITIATOR UE grants the RESPONDER UE permission to transmit unsolicited data, subject to the following general rules:
 - The INITIATOR UE directs the RESPONDER UE which one of the following FUs to use for data transmission:
 - The *Unsolicited Periodic Data Transfer FU* or
 - The *Unsolicited Spontaneous Data Transfer FU*.
 - The desired data are associated with one specified group only.
 - The sequence of the desired data values is the same as in the group definition.
 - The sequence of the desired data values may be a subset of the complete sequence, and the subset may vary from transmission to transmission.
 - The RESPONDER UE may use the specified confirmed, non-confirmed or mixed data service, on a per group basis. The INITIATOR UE recommends which service should be used.
- b) The INITIATOR UE may at any time withdraw the permission to send, forcing the RESPONDER UE to stop the transmission.

7.3.2 Coordination rules

7.3.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Unsolicited Data Transfer FU* are conveyed by one single association. The association shall be permanent, and have the characteristics as specified in 7.3.2.3.1.

7.3.2.2 Relation avec les autres FU

7.3.2.2.1 Invoking FUs

Il est permis à la *Restart Reactivate FU* d'invoquer la *Unsolicited Data Transfer FU*.

7.3.2.2.2 Invoked FUs

Il convient, pour la *Unsolicited Data Transfer FU*, de n'invoquer aucune autre FU⁹²).

7.3.2.2.3 Disrupting FUs

Il est permis aux FU suivantes d'interrompre la *Unsolicited Data Transfer FU*:

- *Permanent Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.3.2.2.4 Disrupted FUs

La *Unsolicited Data Transfer FU* ne doit interrompre aucune autre FU.

7.3.2.3 Invocation

7.3.2.3.1 Conditions préalables

Les FU suivantes doivent avoir été invoquées avant toute invocation de la *Unsolicited Data Transfer FU*:

- *Permanent Association FU*;
- *Group Configuration FU*⁹³).

L'invocation de la *Permanent Association FU* doit toujours être en fonctionnement au moment de l'invocation de la *Unsolicited Data Transfer FU*. L'association maintenue par l'invocation de la *Permanent Association FU* doit également être en fonctionnement (non interrompue temporairement) à ce moment.

La *Permanent Association FU* doit avoir été invoquée pour créer et maintenir l'association à utiliser pour les interactions liées à l'invocation en cours de la *Unsolicited Data Transfer FU*, avec les caractéristiques suivantes:

- Une paire A-suffix, tel que spécifié dans le tableau de 5.2.1.

La *Group Configuration FU* doit avoir été invoquée pour créer et définir le groupe à transmettre, avant l'invocation en cours de la *Unsolicited Data Transfer FU*⁹⁴).

L'invocation de la *Group Configuration FU* de ce groupe ne doit pas être en fonctionnement au moment de l'invocation de la *Unsolicited Data Transfer FU*.

⁹²) L'invocation des FU secondaires *Unsolicited Periodic Data Transfer FU* ou *Unsolicited Spontaneous Data Transfer FU* est à l'initiative du RESPONDER UE.

⁹³) Il est également permis à la Coordinating Function d'invoquer directement la *Group Management FU* et la *Group Definition FU* invoquées par la *Group Configuration FU*.

⁹⁴) L'utilisation de la *Group Configuration FU* n'est cependant pas nécessaire si le groupe est défini dans la RESPONDER AP par un mécanisme non TASE.1 et déclaré dans l'INITIATOR AP par un mécanisme non TASE.1.

7.3.2.2 Relation to other FUs

7.3.2.2.1 Invoking FUs

The *Unsolicited Data Transfer FU* may be invoked by the *Restart Reactivate FU*.

7.3.2.2.2 Invoked FUs

The *Unsolicited Data Transfer FU* should not invoke any other FU⁹²).

7.3.2.2.3 Disrupting FUs

The *Unsolicited Data Transfer FU* may be disrupted by:

- *Permanent Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.3.2.2.4 Disrupted FUs

The *Unsolicited Data Transfer FU* shall not disrupt any other FU.

7.3.2.3 Invocation

7.3.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Unsolicited Data Transfer FU*:

- The *Permanent Association FU*;
- The *Group Configuration FU*⁹³).

The *Permanent Association FU* invocation shall still be running at the time of invocation of the *Unsolicited Data Transfer FU*, and the association maintained by the *Permanent Association FU* invocation shall be running (not temporarily broken) at that time.

The *Permanent Association FU* shall have been invoked to create and maintain the association to be used for the interactions related to the current invocation of the *Unsolicited Data Transfer FU*, with the following characteristics:

- A-suffix pair as specified in the table in 5.2.1.

The *Group Configuration FU* shall have been invoked to create and define the group that is to be transmitted, prior to the current invocation of the *Unsolicited Data Transfer FU*⁹⁴).

The *Group Configuration FU* invocation of that group shall not be running at the time of invocation of the *Unsolicited Data Transfer FU*.

⁹²⁾ The invocation of the secondary FUs *Unsolicited Periodic Data Transfer FU* or *Unsolicited Spontaneous Data Transfer FU* is the responsibility of the RESPONDER UE.

⁹³⁾ Alternatively, the *Group Management FU* and *Group Definition FU*, which are invoked by the *Group Configuration FU*, may have been invoked directly by the Coordinating Function.

⁹⁴⁾ The use of the *Group Configuration FU* is, however, not required if the group is defined in the RESPONDER AP by a non-TASE.1 mechanism and also made known to the INITIATOR AP by a non-TASE.1 mechanism.

L'INITIATOR UE doit faire une proposition en utilisant le paramètre *PreferredService* pour lequel le service de transfert de données est à utiliser par le RESPONDER UE dans la phase de transmission des données ultérieure.

Les services suivants peuvent être utilisés:

SendConfData,
SendNonConfData ou
MixedData.

Il est permis au RESPONDER UE de choisir l'un quelconque de ces services, suivant les décisions locales.

7.3.2.3.2 Restrictions

La *Unsolicited Data Transfer FU* ne doit pas être en fonctionnement lorsque l'une au moins des FU suivantes est en fonctionnement, pour le groupe considéré:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.3.2.3.3 Invoking events

Il est permis d'invoquer la partie INITIATOR de la *Unsolicited Data Transfer FU* par les événements suivants:

- Requête locale par le biais de la Coordinating Function, dont l'origine ne fait partie du domaine d'application du présent rapport technique.
- *Restart Reactivate FU*.

L'invocation de la partie RESPONDER de la *Unsolicited Data Transfer FU* est tentée chaque fois qu'une primitive *UnsolicitedDataTransferMgmt(start)Ind* valide est reçue par le biais d'une association ayant les caractéristiques définies pour la *Unsolicited Data Transfer FU*. Voir la description de la *Permanent Association FU*.

7.3.2.4 Fin

7.3.2.4.1 Orderly termination

Il est admis de déclencher la fin ordonnée de la partie INITIATOR d'une invocation de la *Unsolicited Data Transfer FU* uniquement par les événements suivants:

- Requête locale par le biais de la Coordinating Function. L'origine de cette requête ne fait pas partie du domaine d'application du présent rapport technique.
- Erreur d'encombrement. Voir 7.3.3.2.5.

Il est admis de déclencher la fin ordonnée de la partie RESPONDER d'une invocation de la *Unsolicited Data Transfer FU* uniquement par la réception d'une primitive *UnsolicitedDataTransferMgmt(stop)Ind* valide.

7.3.2.4.2 Disruption

Il est admis d'interrompre la partie INITIATOR et RESPONDER d'une invocation de la *Unsolicited Data Transfer FU* par les événements suivants:

- Interruption d'une autre invocation de FU. Voir 7.3.2.2.3.
- Condition d'erreur fatale rencontrée en fonctionnement. Voir 7.3.3.2.

The INITIATOR UE shall make a proposal using parameter *PreferredService* for which data transfer service to be used by the RESPONDER UE in the subsequent data transmission phase.

The choices are: *SendConfData*,
 SendNonConfData or
 MixedData.

However, the RESPONDER UE may choose any of these services due to local decisions.

7.3.2.3.2 Restrictions

The *Unsolicited Data Transfer FU* shall not be running while at least one of the following FUs are running, for the group involved:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.3.2.3.3 Invoking events

The INITIATOR part of the *Unsolicited Data Transfer FU* may be invoked by:

- Local request via the Coordinating Function, the original source of which is outside the scope of this technical report.
- The *Restart Reactivate FU*.

Invocation of the RESPONDER part of the *Unsolicited Data Transfer FU* is attempted whenever a valid *UnsolicitedDataTransferMgmt(start)Ind* primitive is received via an association with the characteristics as defined for the *Unsolicited Data Transfer FU*: see description of the *Permanent Association FU*.

7.3.2.4 Termination

7.3.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of an *Unsolicited Data Transfer FU* invocation may only be triggered by:

- Local request via the Coordinating Function. The original source of the request is outside the scope of this technical report.
- Congestion error. See 7.3.3.2.5.

Orderly termination of the RESPONDER part of an *Unsolicited Data Transfer FU* invocation may only be triggered by reception of a valid *UnsolicitedDataTransferMgmt(stop)Ind* primitive.

7.3.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of an *Unsolicited Data Transfer FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 7.3.2.2.3.
- A fatal error condition encountered during operation. See 7.3.3.2.

7.3.3 Procédures

7.3.3.1 Primitives de service TASE.1

Le service TASE.1 élémentaire suivant est utilisé par la *Unsolicited Data Transfer FU*:

- *UnsolicitedDataTransferMgnt*

7.3.3.1.1 Sequence

La séquence normale de primitives se répartit en deux phases:

Phase 1: Autorisation d'émettre

INITIATOR UE	TASE.1	RESPONDER UE
<i>UnsolicitedDataTransferMgnt</i> (start,mode)Req	----->	<i>UnsolicitedDataTransferMgnt</i> (start,mode)Ind
<i>UnsolicitedDataTransferMgnt</i> (start,mode)Conf	<-----	<i>UnsolicitedDataTransferMgnt</i> (start,mode)Resp

Où *mode* prend l'une des deux valeurs légales suivantes:

- Periodic: Utiliser la *Unsolicited Periodic Data Transfer FU*;
- Spontaneous: Utiliser la *Unsolicited Spontaneous Data Transfer FU*.

Phase 2: Fin ordonnée: Retrait de l'autorisation d'émettre

INITIATOR UE	TASE.1	RESPONDER UE
<i>UnsolicitedDataTransferMgnt</i> (stop)Req	----->	<i>UnsolicitedDataTransferMgnt</i> (stop)Ind
<i>UnsolicitedDataTransferMgnt</i> (stop)Conf	<-----	<i>UnsolicitedDataTransferMgnt</i> (stop)Resp

7.3.3.1.2 Parameter values

UnsolicitedDataTransferMgnt:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Function	Phase 1: = <i>start</i> Phase 3: = <i>stop</i>	Non applicable
Gtype	Valeur <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> ou <i>Text-message-group</i> , selon le type du groupe considéré, tel que défini dans le CS.	Non applicable
Gnr	Numéro de référence du groupe considéré, tel que défini dans le CS	Non applicable
TransmissionMode	Demande de FU de transmission de données Valeur <i>Periodic</i> ou <i>Spontaneous</i>	Non applicable

7.3.3 Procedures

7.3.3.1 TASE.1 service primitives

The following elementary TASE.1 service is used by the *Unsolicited Data Transfer FU*:

- *UnsolicitedDataTransferMgnt*

7.3.3.1.1 Sequence

The normal sequence of primitives is partitioned into two phases:

Phase 1: Granting permission to send

INITIATOR UE	TASE.1	RESPONDER UE
<i>UnsolicitedDataTransferMgnt</i> (start,mode)Req	----->	<i>UnsolicitedDataTransferMgnt</i> (start,mode)Ind
<i>UnsolicitedDataTransferMgnt</i> (start,mode)Conf	<-----	<i>UnsolicitedDataTransferMgnt</i> (start,mode)Resp

Where *mode* takes one of two legal values;

- Periodic: Use the *Unsolicited Periodic Data Transfer FU*
- Spontaneous: Use the *Unsolicited Spontaneous Data Transfer FU*

Phase 2: Orderly termination: Withdrawing permission to send

INITIATOR UE	TASE.1	RESPONDER UE
<i>UnsolicitedDataTransferMgnt</i> (stop)Req	----->	<i>UnsolicitedDataTransferMgnt</i> (stop)Ind
<i>UnsolicitedDataTransferMgnt</i> (stop)Conf	<-----	<i>UnsolicitedDataTransferMgnt</i> (stop)Resp

7.3.3.1.2 Parameter values

UnsolicitedDataTransferMgnt..:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Function	Phase 1: = <i>start</i> Phase 3: = <i>stop</i>	Not applicable
Gtype	Value <i>Measurand-group</i> , <i>Status-group</i> , <i>Discrete-group</i> , <i>Logical-breaker-status-group</i> , or <i>Text-message-group</i> , depending on the type of the group in question, as defined in the CS.	Not applicable
Gnr	Reference number for the group in question, as defined in the CS	Not applicable
TransmissionMode	Demand of Data transmission FU. Value <i>Periodic</i> or <i>Spontaneous</i>	Not applicable

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
PreferredService	Proposition de la part de l'INITIATOR UE concernant le service de transfert de données à utiliser par le RESPONDER ⁹⁵⁾ Valeur <i>ConfData</i> , <i>NonConfData</i> ou <i>MixedData</i>	Non applicable
Result	(Non applicable)	Action effectuée comme spécifié. = <i>result-ok</i> Action non effectuée en raison d'une condition d'erreur: Autre valeur. Voir 7.3.3.2. (si <i>Function = start</i> , ceci signifie que la partie RESPONDER de la <i>Unsolicited Data Transfer FU</i> n'a pas été invoquée pour le groupe considéré).

7.3.3.2 Traitement des erreurs

7.3.3.2.1 FU disruption

Interruption par la *Permanent Association FU*:

L'interruption de la partie INITIATOR et de la partie RESPONDER de l'invocation en cours de la *Unsolicited Data Transfer FU* doit être déclenchée localement, comme faisant partie du traitement des primitives *TASEAbortInd* entrantes dans la partie INITIATOR et dans la partie RESPONDER de l'invocation de la *Permanent Association FU* traitant l'association sur laquelle fonctionne la *Unsolicited Data Transfer FU*.

Les deux parties de l'invocation en cours de la *Unsolicited Data Transfer FU* doivent se terminer correctement, aucune des deux parties ne devant tenter d'émettre une primitive associée à la fin de l'invocation elle-même⁹⁶⁾.

Une fois l'association terminée, la *Permanent Association FU* passe dans un état dans lequel il lui est permis de signaler le code de redémarrage *Restart*, *spontaneous management lost*, dès qu'une association ayant les caractéristiques permettant le *Unsolicited Data Transfer* est établie entre la même paire INITIATOR/RESPONDER UE⁹⁷⁾.

7.3.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Unsolicited Data Transfer FU* n'est pas présente dans un INITIATOR UE:

Les requêtes d'invocation sont toujours générées localement. Voir 7.3.2.3. En conséquence, ce type d'erreur est traité localement et ne fait pas partie du domaine d'application de le présent rapport technique.

⁹⁵⁾ L'INITIATOR UE recommande au RESPONDER UE le service Data Transfer à utiliser. Il est cependant permis au RESPONDER UE de choisir tout autre service de transmission de données, les décisions pouvant être prises localement.

⁹⁶⁾ Les procédures de nettoyage locales ne sont pas spécifiées par le présent rapport technique.

⁹⁷⁾ Il est permis au RESPONDER UE d'être programmé de façon à toujours signaler le code de redémarrage "*Restart*, *spontaneous management lost*" lors de la création d'associations pour le *Unsolicited Data Transfer*, quel que soit l'historique précédent.

Parameter	Req (INITIATOR)	Resp (RESPONDER)
PreferredService	Proposal from INITIATOR UE for the data transfer service to be used by RESPONDER ⁹⁵⁾ Value <i>ConfData</i> , <i>NonConfData</i> or <i>MixedData</i>	Not applicable
Result	(Not applicable)	Action performed as specified. = <i>result-ok</i> Action not performed, due to error condition: Other value. See 7.3.3.2. (If Function = <i>start</i> , this means that the RESPONDER part of the <i>Unsolicited Data Transfer FU</i> has not been invoked for the group in question).

7.3.3.2 Error handling

7.3.3.2.1 FU disruption

Disruption by the *Permanent Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Unsolicited Data Transfer FU* shall be triggered locally, as a part of the handling of incoming *TASEAbortInd* primitives in both the INITIATOR part and the RESPONDER part of the *Permanent Association FU* invocation handling the association on which the *Unsolicited Data Transfer FU* is running.

Both parts of the current invocation of the *Unsolicited Data Transfer FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself⁹⁶⁾.

Following the termination, the *Permanent Association FU* will enter a state in which it may signal *Restart*, *spontaneous management lost*, the next time an association with the characteristics for *Unsolicited Data Transfer* is established between the same INITIATOR and RESPONDER UE pair⁹⁷⁾.

7.3.3.2.2 Illegal invocation attempt

FU not present:

If the *Unsolicited Data Transfer FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see 7.3.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this technical report.

⁹⁵⁾ The INITIATOR UE recommends which Data Transfer service should be used by the RESPONDER UE. However, the RESPONDER UE may choose any service for data transmission due to local decisions.

⁹⁶⁾ Local clean-up procedures are not specified by this technical report.

⁹⁷⁾ The RESPONDER UE may be programmed to always signal the restart code "*Restart, spontaneous management lost*" during creation of associations for *Unsolicited Data Transfer*, regardless of previous history.

Si la *Unsolicited Data Transfer FU* n'est pas présente dans un RESPONDER UE:

Le RESPONDER User Element doit répondre aux tentatives d'activation en:

- émettant une primitive *UnsolicitedDataTransferMgnt(start)Resp* avec Result = *remote-service-user-unavailable*

FU présente:

La réinvocation de la *Unsolicited Data Transfer FU* est traitée comme une invocation légale, mais aucune FU nouvelle n'est démarrée et des valeurs d'attribut antérieures doivent être utilisées.

7.3.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>UnsolicitedDataTransferMgnt(start)Conf</i>	<i>UnsolicitedDataTransferMgnt(stop)Conf</i>
FU non en fonctionnement	Ignorer, ou indication d'erreur locale/connexion	Ignorer, ou indication d'erreur locale/connexion
FU en fonctionnement, en attente de <i>UnsolicitedDataTransferMgnt(start)Conf</i>	(Normal)	(Impossible)
En attente de <i>UnsolicitedDataTransferMgnt(stop)Conf</i>	(Impossible)	(Normal)

Partie RESPONDER:

Etat	<i>UnsolicitedDataTransferMgnt(start)Ind</i>	<i>UnsolicitedDataTransferMgnt(stop)Ind</i>
FU non en fonctionnement	(Normal)	Ignorer, ou: Emettre une primitive <i>UnsolicitedDataTransferMgnt(stop)Resp</i> avec Result = <i>result-ok</i> , sans terminer aucune invocation de <i>Unsolicited Data Transfer FU</i> dans le RESPONDER UE.

7.3.3.2.4 Timing errors

Erreur dans la partie INITIATOR:

Aucune.

Erreur dans la partie RESPONDER:

Erreur	Réaction de la part de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à la primitive <i>UnsolicitedDataTransferMgnt(start)Ind</i>	Dans le RESPONDER: Erreur locale lors d'une éventuelle tentative d'émission de la primitive <i>UnsolicitedDataTransferMgnt(start)Resp</i> Dans l'INITIATOR: <i>UnsolicitedDataTransferMgnt(start)Conf</i> avec Result = <i>remote-service-user-unavailable</i>.	Partie RESPONDER: Terminer l'invocation de la FU localement. Partie INITIATOR: Terminer l'invocation de la FU localement

If the *Unsolicited Data Transfer FU* is not present in a RESPONDER UE:

The RESPONDER User Element shall respond to activation attempts by:

- Issuing an *UnsolicitedDataTransferMgmt(start)Resp* primitive with Result = *remote-service-user-unavailable*

FU present:

Re-invoking the *Unsolicited Data Transfer FU* is treated as a legal invocation, but no new FU is started and earlier attribute values shall be used.

7.3.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>UnsolicitedDataTransferMgmt(start)Conf</i>	<i>UnsolicitedDataTransferMgmt(stop)Conf</i>
FU not running	Ignore, or local error indication/logging	Ignore, or local error indication/logging
FU running, waiting for <i>UnsolicitedDataTransferMgmt(start)Conf</i>	(Normal)	(Impossible)
Waiting for <i>UnsolicitedDataTransferMgmt(stop)Conf</i>	(Impossible)	(Normal)

RESPONDER part:

State	<i>UnsolicitedDataTransferMgmt(start)Ind</i>	<i>UnsolicitedDataTransferMgmt(stop)Ind</i>
FU not running	(Normal)	Ignore, or: Issue an <i>UnsolicitedDataTransferMgmt(stop)Resp</i> primitive with Result = <i>result-ok</i> , without terminating any <i>Unsolicited Data Transfer FU</i> invocation in the RESPONDER UE.

7.3.3.2.4 Timing errors

Error in INITIATOR part:

None.

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>UnsolicitedDataTransferMgmt(start)Ind</i>	In RESPONDER: Local error from eventual attempt at issuing <i>UnsolicitedDataTransferMgmt(start)Resp</i> In INITIATOR: <i>UnsolicitedDataTransferMgmt(start)Conf</i> , with Result = <i>remote-service-user-unavailable</i> .	RESPONDER part: Terminate FU invocation locally. INITIATOR part: Terminate FU invocation locally.

Erreur	Réaction de la part de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à la primitive <i>UnsolicitedDataTransferMgnt(stop)Ind</i>	Dans le RESPONDER: Erreur locale lors d'une éventuelle tentative d'émission de la primitive <i>UnsolicitedDataTransferMgnt.(stop)Resp</i> Dans l'INITIATOR: <i>Unsolicited Data Transfer Mgnt (stop)Conf</i> avec Result = <i>misbehaviour-of-remote-service-user</i>.	Partie RESPONDER: Terminer l'invocation de la FU localement. Partie INITIATOR: Terminer l'invocation de la FU localement

7.3.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsque ce type d'erreur intervient avec une tentative de *UnsolicitedDataTransferMgnt(start)Req*:

- terminer l'invocation de la FU localement.

Lorsque ce type d'erreur intervient avec une tentative de *UnsolicitedDataTransferMgnt(stop)Req*:

- ignorer ou prévenir l'opérateur.

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsque ce type d'erreur intervient avec une tentative de *UnsolicitedDataTransferMgnt(start)Resp*:

- terminer l'invocation de la FU localement. (La partie INITIATOR de la FU se terminera finalement après l'expiration du délai du fournisseur de TASE.1.)

Lorsque ce type d'erreur intervient avec une tentative de *UnsolicitedDataTransferMgnt(stop)Resp*:

- terminer l'invocation de la FU localement. (La partie INITIATOR de la FU se terminera finalement après l'expiration du délai du fournisseur de TASE.1.)

Voir 4.2.1.

7.3.3.2.6 TASE.1 service primitive parameter errors

Dans le contexte de cette FU, le paramètre Gnr est valide pour toutes les primitives, sauf la primitive *UnsolicitedDataTransferMgnt(start)Ind* et *uniquement si la FU est en cours de fonctionnement*⁹⁸⁾ pour ce groupe. Ce paramètre permet d'identifier l'invocation de la FU vers laquelle la primitive entrante doit être dirigée.

⁹⁸⁾ Le cas de la FU non invoquée pour le groupe doit être traité en dehors de la FU (par la Coordinating Function): La primitive de réponse naturelle doit être émise, avec le paramètre Result = *gnr-out-of-range*.

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>UnsolicitedDataTransferMgmt</i> (<i>stop</i>) <i>Ind</i>	In RESPONDER: Local error from eventual attempt at issuing <i>UnsolicitedDataTransfer</i> <i>Mgmt(stop)Resp</i> In INITIATOR: <i>UnsolicitedDataTransferMgmt.</i> (<i>stop</i>) <i>Conf</i> , with Result = <i>misbehaviour-of-remote-service-</i> <i>user</i> .	RESPONDER part: Terminate FU invocation locally. INITIATOR part: Terminate FU invocation locally.

7.3.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with an *UnsolicitedDataTransferMgmt(start)Req* attempt:

- terminate FU invocation locally.

When occurring with an *UnsolicitedDataTransferMgmt(stop)Req* attempt:

- ignore, or notify operator.

See 4.2.1.

RESPONDER part of the FU:

When occurring with an *UnsolicitedDataTransferMgmt(start)Resp* attempt:

- terminate FU invocation locally. (INITIATOR part of FU will eventually be terminated after timeout in the TASE.1 provider.)

When occurring with an *UnsolicitedDataTransferMgmt(stop)Resp* attempt:

- terminate FU invocation locally. (INITIATOR part of FU will eventually be terminated after timeout in the TASE.1 provider.)

See 4.2.1.

7.3.3.2.6 TASE.1 service primitive parameter errors

For all primitives except the *UnsolicitedDataTransferMgmt(start)Ind* primitive, the parameter Gnr will always be valid within the context of this FU, *provided the FU is running*⁹⁸⁾ for that group. It serves as identification of the FU invocation to which the incoming primitive shall be directed.

⁹⁸⁾ The case of the FU not being invoked for the group shall be handled outside the FU (by the Coordinating Function): the natural replying primitive shall be issued, with parameter Result = *gnr-out-of-range*.

Erreurs dans la primitive *UnsolicitedDataTransferMgmt(start)Ind* (détectées par la partie RESPONDER):

Erreur	Action dans la partie RESPONDER de la FU
Valeur Gtype non égale à la valeur de l'attribut Group type dans le CS(R) pour le groupe n° Gnr. (le cas de l'absence d'attribut Group. Gnr est examiné ci-dessous.)	Emettre la primitive <i>UnsolicitedDataTransferMgmt(start)Resp</i> avec Result = <i>gtype-out-of-range</i> et les valeurs de Group-number et Group-type identiques à celles de la primitive <i>Ind</i> correspondante. Ne pas invoquer la FU.
Valeur de Gnr illégale ou groupe n° Gnr inexistant dans le CS(R).	Emettre la primitive <i>UnsolicitedDataTransferMgmt(start)Resp</i> avec Result = <i>gnr-out-of-range</i> et les valeurs de Group-number et Group-type identiques à celles de la primitive <i>Ind</i> correspondante. Ne pas invoquer la FU.

Erreurs dans la primitive *UnsolicitedDataTransferMgmt(stop)Ind* (détectées par la partie RESPONDER):

Erreur	Action dans la partie RESPONDER de la FU
Valeur Gtype différente de la valeur présente dans le CS(R) pour le groupe associé à l'invocation de FU en cours.	Emettre la primitive <i>UnsolicitedDataTransferMgmt(stop)Resp</i> avec Result = <i>gtype-out-of-range</i> et les valeurs de Gnr et Gtype identiques à celles de la primitive <i>Ind</i> correspondante. Ne pas terminer la FU.

Erreurs dans la primitive *UnsolicitedDataTransferMgmt(start)Conf* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Absence de correspondance entre la valeur de Gtype dans la primitive et la valeur de Gtype dans la primitive <i>UnsolicitedDataTransferMgmt(start)Req</i> correspondante, avec Result = <i>result-ok</i> .	Déclencher une fin ordonnée de l'invocation de FU en cours, en émettant une primitive <i>UnsolicitedDataTransferMgmt(stop)Req</i> pour le groupe considéré, avec la valeur de Gtype identique à celle utilisée dans la primitive <i>UnsolicitedDataTransferMgmt(start)Req</i> correspondante.
Result >< <i>result-ok</i>	Terminer l'invocation de la FU localement

Erreur dans la primitive *UnsolicitedDataTransferMgmt(stop)Conf* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Absence de correspondance entre la valeur de Gtype dans la primitive et la valeur de Gtype dans la primitive <i>UnsolicitedDataTransferMgmt(stop)Req</i> correspondante, avec Result = <i>result-ok</i> .	Déclencher une fin brutale de l'association
Result >< <i>result-ok</i>	Déclencher une fin brutale de l'association

7.4 Unsolicited Periodic Data Transfer FU

Type: Secondary.

7.4.1 Fonction

La *Unsolicited Periodic Data Transfer FU* est de type Secondary, ce qui signifie que l'invocation de la FU est déclenchée exclusivement par le RESPONDER UE. Il est permis à la partie RESPONDER de la *Unsolicited Periodic Data Transfer FU* d'utiliser l'une des primitives de services suivantes:

Errors in the *UnsolicitedDataTransferMgmt(start)Ind* primitive (detected by the RESPONDER part):

Error	Action in RESPONDER part of FU
Gtype value not equal to value of attribute Group type in CS(R) for Group No. Gnr. (The case of non-existing Group. Gnr is considered below.)	Issue <i>UnsolicitedDataTransferMgmt(start)Resp</i> with Result = <i>gtype-out-of-range</i> , and the values of Group-number and Group-type as in the corresponding <i>Ind</i> . Do not invoke the FU.
Value of Gnr is illegal, or group No. Gnr does not exist in the CS(R).	Issue <i>UnsolicitedDataTransferMgmt(start)Resp</i> with Result = <i>gnr-out-of-range</i> , and the values of Group-number and Group-type as in the corresponding <i>Ind</i> . Do not invoke the FU.

Errors in the *UnsolicitedDataTransferMgmt(stop)Ind* primitive (detected by the RESPONDER part):

Error	Action in RESPONDER part of FU
Gtype value different from the value in CS(R) for the group associated with the current FU invocation	Issue <i>UnsolicitedDataTransferMgmt(stop)Resp</i> with Result = <i>gtype-out-of-range</i> , and the values of Gnr and Gtype as in the corresponding <i>Ind</i> . Do not terminate the FU.

Errors in the *UnsolicitedDataTransferMgmt(start)Conf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Mismatch between Gtype in primitive and Gtype in corresponding <i>UnsolicitedDataTransferMgmt(start)Req</i> , and Result = <i>result-ok</i> .	Trigger Orderly Termination of the current FU invocation, issuing <i>UnsolicitedDataTransferMgmt(stop)Req</i> for the group in question, with value of Gtype as in the corresponding <i>UnsolicitedDataTransferMgmt(start)Req</i> .
Result >< <i>result-ok</i>	Terminate the FU invocation locally

Errors in the *UnsolicitedDataTransferMgmt(stop)Conf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Mismatch between Gtype in primitive and Gtype in corresponding <i>UnsolicitedDataTransferMgmt(stop)Req</i> , and Result = <i>result-ok</i> .	Trigger abrupt termination of supporting association
Result >< <i>result-ok</i>	Trigger abrupt termination of supporting association

7.4 Unsolicited Periodic Data Transfer FU

Type: Secondary.

7.4.1 Function

The *Unsolicited Periodic Data Transfer FU* is of type Secondary, meaning that invocation of the FU is triggered exclusively by the RESPONDER UE. The RESPONDER part of an *Unsolicited Periodic Data Transfer FU* may use one of the following service primitives:

- *SendConfData*;
- *SendNonConfData*;
- *MixedData*.

Le RESPONDER UE choisit le service à utiliser. Il est cependant permis à l'INITIATOR UE de proposer un service en utilisant le paramètre *PreferredService* lorsqu'il invoque la *Unsolicited Data Transfer FU*.

Lorsque la FU est invoquée, l'heure de l'invocation ainsi que la taille et le contenu des données sont déterminés par le RESPONDER UE, selon les règles suivantes:

- La *Unsolicited Data Transfer FU* doit être en fonctionnement pour le groupe pour lequel la *Unsolicited Periodic Data Transfer FU* est à invoquer.
- La séquence des valeurs de données de toute transmission est identique à celle de la définition du groupe.
- Pour les groupes de type *Text-message-group*, chaque invocation de FU doit contenir la valeur de données d'un objet et d'un seul.
- Il est permis au RESPONDER UE d'invoquer la *Unsolicited Periodic Data Transfer FU* autant de fois qu'il le souhaite, jusqu'à ce que l'INITIATOR UE retire l'autorisation d'émettre. Cette action est effectuée en terminant de façon ordonnée la *Unsolicited Data Transfer FU*.
- Si le service *MixedData* est utilisé, les données sont associées à un groupe TASE.1 spécifié et sont soumises uniquement aux deux règles suivantes:
 - a) L'intervalle de temps entre les valeurs ne doit pas être trop grand. Voir annexe A de la CEI 60870-6-501 traitant de la zone «Delta Time» du paramètre *MixedData*.
 - b) La transmission de groupes de type *Text-message-group* n'est pas autorisée.

7.4.2 Règles de coordination

7.4.2.1 Utilisation d'une association

Toutes les interactions de TASE.1 faisant partie d'une invocation de la *Unsolicited Periodic Data Transfer FU* sont véhiculées par une association unique. L'association doit être permanente et avoir les caractéristiques spécifiées en 7.4.2.3.1.

7.4.2.2 Relation avec les autres FU

7.4.2.2.1 Invoking FUs

La *Unsolicited Periodic Data Transfer FU* est invoquée à la demande de la Coordination Function locale du RESPONDER UE.

7.4.2.2.2 Invoked FUs

La *Unsolicited Periodic Data Transfer FU* ne doit invoquer aucune autre FU.

7.4.2.2.3 Disrupting FUs

Il est permis aux FU suivantes d'interrompre la *Unsolicited Periodic Data Transfer FU*:

- *Permanent Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

- *SendConfData*,
- *SendNonConfData*,
- *MixedData*.

The decision of which service to use is done by the RESPONDER UE. The INITIATOR UE, however, may desire a service by using the parameter *PreferredService* when invoking the *Unsolicited Data Transfer FU*.

When invoked, the time of invocation, as well as the size and contents of the data part are determined by the RESPONDER UE, subject to the following rules:

- The *Unsolicited Data Transfer FU* must currently be running for the group for which the *Unsolicited Periodic Data Transfer FU* is to be invoked.
- The sequence of the data values in any transmission is the same as in the group definition.
- For groups of type *Text-message-group*, each FU invocation shall contain the data value of exactly one object.
- The RESPONDER UE may invoke the *Unsolicited Periodic Data Transfer FU* any number of times, until the INITIATOR UE withdraws the permission to send. This is done by orderly terminating the *Unsolicited Data Transfer FU*.
- If the *MixedData* service is used, the data is associated with one specified TASE.1 group, subject only to two limitations:
 - a) The time span between values must not be too great, see annex A of IEC 60870-6-501, the “Delta Time” field of the *MixedData* parameter.
 - b) Transmitting groups of type *Text-message-group* is not allowed.

7.4.2 Coordination rules

7.4.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Unsolicited Periodic Data Transfer FU* are conveyed by one single association. The association shall be permanent, and have the characteristics as specified in 7.4.2.3.1.

7.4.2.2 Relation to other FUs

7.4.2.2.1 Invoking FUs

The *Unsolicited Periodic Data Transfer FU* is invoked by request from the local Coordination Function in the RESPONDER UE.

7.4.2.2.2 Invoked FUs

The *Unsolicited Periodic Data Transfer FU* shall not invoke any other FU.

7.4.2.2.3 Disrupting FUs

The *Unsolicited Periodic Data Transfer FU* may be disrupted by:

- *Permanent Association FU*
- *Group Management FU*
- *Group Definition FU*
- *Restart Reconfigure FU*.

7.4.2.2.4 Disrupted FUs

La *Unsolicited Periodic Data Transfer FU* ne doit interrompre aucune autre FU.

7.4.2.3 Invocation

7.4.2.3.1 Conditions préalables

Les FU suivantes doivent avoir été invoquées avant toute invocation de la *Unsolicited Periodic Data Transfer FU*:

- *Permanent Association FU*;
- *Group Configuration FU* ⁹⁹⁾;
- *Unsolicited Data Transfer FU*.

L'invocation de la *Permanent Association FU* doit encore être en cours de fonctionnement au moment de l'invocation de la *Unsolicited Periodic Data Transfer FU*. L'association maintenue par l'invocation de la *Permanent Association FU* doit également être en cours de fonctionnement (et pas interrompue temporairement) au moment de l'invocation.

La *Permanent Association FU* doit avoir été invoquée pour pouvoir créer et maintenir l'association à utiliser pour les interactions liées à l'invocation en cours de la *Unsolicited Periodic Data Transfer FU*, en utilisant les caractéristiques suivantes:

- Paire A-suffix telle que spécifié dans le tableau de 5.2.1.

La *Group Configuration FU* doit avoir été invoquée pour pouvoir définir et configurer le groupe à transmettre, avant l'invocation en cours de la *Unsolicited Periodic Data Transfer FU* ¹⁰⁰⁾.

L'invocation de la *Group Configuration FU* de ce groupe ne doit pas être en cours de fonctionnement au moment de l'invocation de la *Unsolicited Periodic Data Transfer FU*.

La *Unsolicited Data Transfer FU* doit encore être en fonctionnement au moment de l'invocation de la *Unsolicited Periodic Data Transfer FU*.

7.4.2.3.2 Restrictions

La *Unsolicited Periodic Data Transfer FU* ne doit pas être invoquée tant que l'une au moins des FU suivantes est en fonctionnement, pour le groupe considéré:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.4.2.3.3 Invoking events

La partie INITIATOR de la *Unsolicited Periodic Data Transfer FU* est toujours invoquée par la réception d'une primitive de service *SendConfDataInd*, *SendNonConfDataInd* ou *MixedDataInd* par le biais d'une association ayant les caractéristiques définies pour la *Unsolicited Periodic Data Transfer FU*.

⁹⁹⁾ Il est également permis à la Coordinating Function d'invoquer directement la *Group Management FU* et la *Group Definition FU*, elles-mêmes invoquées par la *Group Configuration FU*.

¹⁰⁰⁾ L'utilisation de la *Group Configuration FU* ou de la *Group Management FU* et de la *Group Definition FU* n'est cependant pas nécessaire si le groupe est défini dans la RESPONDER AP par un mécanisme non normalisé et est également déclaré dans l'INITIATOR AP par un mécanisme non normalisé.

7.4.2.2.4 Disrupted FUs

The *Unsolicited Periodic Data Transfer FU* shall not disrupt any other FU.

7.4.2.3 Invocation

7.4.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Unsolicited Periodic Data Transfer FU*:

- *Permanent Association FU*
- *Group Configuration FU* ⁹⁹⁾
- *Unsolicited Data Transfer FU*

The *Permanent Association FU* invocation shall still be running at the time of invocation of the *Unsolicited Periodic Data Transfer FU*, and the association maintained by the *Permanent Association FU* invocation shall be running (not temporarily broken) at that time.

The *Permanent Association FU* shall have been invoked to create and maintain the association to be used for the interactions related to the current invocation of the *Unsolicited Periodic Data Transfer FU*, with the following characteristics:

- A-suffix pair as specified in the table in 5.2.1.

The *Group Configuration FU* shall have been invoked to define and configure the group that is to be transmitted, prior to the current invocation of the *Unsolicited Periodic Data Transfer FU* ¹⁰⁰⁾.

The *Group Configuration FU* invocation of that group shall not be running at the time of invocation of the *Unsolicited Periodic Data Transfer FU*.

The *Unsolicited Data Transfer FU* shall still be running at the time of invocation of the *Unsolicited Periodic Data Transfer FU*.

7.4.2.3.2 Restrictions

The *Unsolicited Periodic Data Transfer FU* must not be invoked while at least one of the following FUs are running, for the group involved:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.4.2.3.3 Invoking events

The INITIATOR part of the *Unsolicited Periodic Data Transfer FU* is always invoked by reception of a *SendConfDataInd*, *SendNonConfDataInd* or *MixedDataInd* service primitive via an association with the characteristics as defined for the *Unsolicited Periodic Data Transfer FU*.

⁹⁹⁾ Alternatively, the *Group Management FU* and *Group Definition FU*, which are invoked by the *Group Configuration FU*, may have been invoked directly by the Coordinating Function.

¹⁰⁰⁾ The use of the *Group Configuration FU* or *Group Management FU* and *Group Definition FU* is, however, not required if the group is defined in the RESPONDER AP by a non-standardized mechanism and also made known to the INITIATOR AP by a non-standardized mechanism.

L'invocation de la partie RESPONDER de la *Unsolicited Periodic Data Transfer FU* est effectuée par un événement local après la réception d'une primitive *UnsolicitedDataTransferMgmtInd(start, periodic)* valide.

7.4.2.4 Fin

7.4.2.4.1 Orderly termination

Il est permis aux événements suivants de déclencher la fin ordonnée de la partie RESPONDER d'une invocation de la *Unsolicited Periodic Data Transfer FU*:

- Requête locale par le biais de la Coordinating Function. L'origine de la requête ne fait pas partie du domaine d'application du présent rapport technique.
- Réception d'une primitive *UnsolicitedDataTransferMgmtInd(stop, periodic)* valide.
- Erreur d'encombrement. Voir 7.4.3.2.5.

7.4.2.4.2 Disruption

Il est permis aux événements suivants d'interrompre la partie INITIATOR et RESPONDER d'une invocation de la *Unsolicited Periodic Data Transfer FU*:

- Interruption d'une autre invocation de FU. Voir 7.4.2.2.3.
- Condition d'erreur fatale rencontrée en fonctionnement. Voir 7.4.3.2.

7.4.3 Procédures

7.4.3.1 Primitives de service de TASE.1

Les services de TASE.1 élémentaires suivants sont utilisés par la *Unsolicited Periodic Data Transfer FU*:

- *SendConfData*;
- *SendNonConfData*;
- *MixedData*.

Le choix du service est effectué localement par le RESPONDER UE et ne fait pas partie du domaine d'application du présent rapport technique. L'INITIATOR UE doit faire une proposition de primitive de service à utiliser par le paramètre *PreferredService*, qui appartient à la primitive *UnsolicitedDataTransferMgmtInd(Start, Periodic, <PreferredService>)*, où *PreferredService* peut prendre l'une des trois valeurs suivantes:

- *SendConfData*;
- *SendNonConfData*;
- *SendMixedData*.

Il est permis au RESPONDER UE de choisir indépendamment entre les primitives de service disponibles chaque fois que la *Unsolicited Periodic Data Transfer FU* est invoquée.

7.4.3.1.1 Sequence

La séquence normale de primitives apparaît différemment selon le type de primitives de service choisies parmi les trois disponibles:

Invocation of the RESPONDER part of the *Unsolicited Periodic Data Transfer FU* is done by a local event after a valid *UnsolicitedDataTransferMgmtInd(start,periodic)* primitive is received.

7.4.2.4 Termination

7.4.2.4.1 Orderly termination

Orderly termination of the RESPONDER part of an *Unsolicited Periodic Data Transfer FU* invocation may only be triggered by:

- Local request via the Coordinating Function. The original source of the request is outside the scope of this technical report.
- Reception of a valid *UnsolicitedDataTransferMgmtInd(stop,periodic)* primitive.
- Congestion error. See 7.4.3.2.5.

7.4.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Unsolicited Periodic Data Transfer FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 7.4.2.2.3.
- A fatal error condition encountered during operation. See 7.4.3.2.

7.4.3 Procedures

7.4.3.1 TASE.1 service primitives

The following elementary TASE.1 services are used by the *UnsolicitedPeriodicDataTransferFU*:

- *SendConfData*;
- *SendNonConfData*;
- *MixedData*;

the use of which is a local decision of the RESPONDER UE, outside the scope of this technical report. The INITIATOR UE shall suggest which service primitive to use by the parameter *PreferredService*, belonging to the *UnsolicitedDataTransferMgmtInd(Start,Periodic, <Preferred Service>)* primitive, where *PreferredService* can take one of three values:

- *SendConfData*.
- *SendNonConfData*,
- *SendMixedData*.

The RESPONDER UE may choose independently between the available service primitives, each time the *Unsolicited Periodic Data Transfer FU* is invoked.

7.4.3.1.1 Sequence

The normal sequence of primitives is shown differently depending on which of the three service primitives are chosen:

Cas n°1: Utilisation de la primitive *SendConfData*

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendConfData(spont.)Ind</i>	<-----	<i>SendConfData(spont.)Req</i>
<i>SendConfData(spont.)Resp</i>	----->	<i>SendConfData(spont.)Conf</i>
The sequence is repeated until the FU is ended.		

Cas n°2: Utilisation de la primitive *SendNonConfData*

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendNonConfData (spont.)Ind</i>	<-----	<i>SendNonConfData(spont.)Req</i>
<i>SendNonConfData(spont.)Ind</i>	<-----	<i>SendNonConfData(spont.)Req</i>

Cas n°3: Utilisation de la primitive *MixedData*

INITIATOR UE	TASE.1	RESPONDER UE
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>

Les règles suivantes sont applicables:

- a) Le RESPONDER doit garder constante la longueur de la période, ou intervalle de temps entre les primitives *SendNonConfData(spont.)Req*, *SendConfData(spont.)Req* ou *MixedDataReq* successives tout au long de l'invocation de la FU. Le RESPONDER UE doit déterminer la longueur de la période à partir de la valeur de l'attribut Priority class du groupe¹⁰¹⁾ pour lequel la FU est invoquée:
 - Si Priority class = 0, le RESPONDER UE doit toujours envoyer les données périodiquement, mais il est permis de choisir la longueur de la période en fonction des capacités du système RESPONDER.
 - Pour toutes les autres valeurs de Priority class, la valeur doit être interprétée dans le RESPONDER UE comme une requête pour une longueur de période définie par une entrée d'une table des longueurs de période spécifique du RESPONDER. Cette table est différente selon les implémentations mais doit être conforme aux règles suivantes:
 - 1) Le déplacement vers des valeurs plus petites de Priority class doit correspondre au déplacement vers des longueurs de périodes plus courtes.
 - 2) La valeur Priority class 1 doit correspondre à la plus courte longueur de période que l'implémentation du RESPONDER UE peut supporter.

Il est permis à la table des longueurs de périodes de contenir moins de 15 entrées, en réunissant plusieurs Priority class en une seule. La table doit cependant contenir au moins deux entrées, si le mécanisme est supporté par un RESPONDER UE donné. Ces deux entrées doivent correspondre aux valeurs 1 et 15, respectivement.

- b) Les règles établies en 7.4.1 s'appliquent également ici.

¹⁰¹⁾ Si un objet appartient à plusieurs groupes, chaque occurrence des données de la queue sortante doit être traitée indépendamment, en fonction des valeurs de Priority class individuelles.

Alternative 1: *SendConfData* primitive is used.

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendConfData(spont)Ind</i>	<-----	<i>SendConfData(spont)Req</i>
<i>SendConfData(spont)Resp</i>	----->	<i>SendConfData(spont)Conf</i>
	.	
	The sequence is repeated until the FU is ended.	

Alternative 2: *SendNonConfData* primitive is used.

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendNonConfData(spont)Ind</i>	<-----	<i>SendNonConfData(spont)Req</i>
<i>SendNonConfData(spont)Ind</i>	<-----	<i>SendNonConfData(spont)Req</i>

Alternative 3: *MixedData* primitive is used.

INITIATOR UE	TASE.1	RESPONDER UE
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>

The following rules apply:

a) The period length, or time span between successive *SendNonConfData(spont)Req*, *SendConfData(spont)Req* or *MixedDataReq* primitives, shall be kept constant by the RESPONDER during the lifetime of the FU invocation. The RESPONDER UE shall determine period length based on the value of attribute Priority class for the group¹⁰¹⁾ for which the FU is invoked:

- If Priority class = 0, the RESPONDER UE shall still send data periodically, but the periodic length may be chosen based on the capabilities of the RESPONDER system.
- For all other values of Priority class, the value shall be interpreted in the RESPONDER UE as a request for a period length defined by an entry in a RESPONDER specific table of supported period lengths. This table is implementation dependent, but shall conform to the following rules:
 - 1) Moving towards smaller Priority class values shall correspond to moving to shorter period lengths.
 - 2) Priority class value 1 shall correspond to the shortest period length that the RESPONDER UE implementation can support.

The number of entries in the period length table may be less than 15, effectively collapsing a number of Priority class values into one. However, the table shall contain at least two entries, if the mechanism is at all supported by a given RESPONDER UE. These two entries shall then correspond to values 1 and 15, respectively.

b) The rules stated in 7.4.1 also apply here.

¹⁰¹⁾ If an object belongs to more than one group, each occurrence of the data in the outgoing queue shall be handled independently, according to their individual Priority class values.

7.4.3.1.2 Parameter values

Send(Non)ConfDataReq (RESPONDER):

Paramètre	Req
Gtype	Copie de la valeur contenue dans la primitive <i>UnsolicitedDataTransferMgmt(start)Ind</i> de la FU Primary (<i>Unsolicited Data Transfer FU</i>).
Gnr	Copie de la valeur contenue dans la primitive <i>UnsolicitedDataTransferMgmt(start) Ind</i> de la FU Primary.
Index1 ¹⁰²⁾	Doit être >0; Egal au plus faible Object Number, selon le CS(R), de la plage d'Object Numbers dont les valeurs de données sont contenues dans le paramètre Data (voir ci-dessous) Doit être égal à Index2 si Gtype = <i>Text-message-group</i> ¹⁰³⁾
Index2	Doit être >= valeur de Index1: Egal à l'Object number le plus élevé, selon le CS(R), de la plage d'Object numbers dont les valeurs de données sont contenues dans le paramètre Data (voir ci-dessous). Doit être égal à Index1 si Gtype = <i>Text-message-group</i> .
T	Date et heure de l'ensemble du jeu de valeurs contenues dans le paramètre Data (voir ci-dessous) et déterminées par le RESPONDER UE. Il est conseillé d'utiliser l'heure UTC.
Data	Données réelles transférées. La structure est définie dans l'annexe A de la CEI 60870-6-501.
Uniquement pour <i>SendNonConfDataReq</i> : Result	= <i>result-ok</i>

SendConfDataResp (INITIATOR):

Paramètre	Resp
Result	Si aucune erreur n'est détectée par l'INITIATOR UE: = <i>result-ok</i> . Si une erreur est détectée par l'INITIATOR UE: Autre valeur. Voir 7.4.3.2.

MixedDataReq (RESPONDER):

Paramètre	Req
T	Date à laquelle toutes les valeurs de temps codées dans le paramètre Data sont liées. Il est conseillé d'utiliser l'heure UTC.
Data	Pour chaque valeur de données: – Group number: Valeur de l'attribut Group number – indexnumber: Valeur de l'attribut Object number – Delta Time: Heure, par rapport au paramètre T – Data: Code de qualité et valeur. Le codage est décrit dans l'annexe A de la CEI 60870-6-501.

¹⁰²⁾ Le nombre de valeurs qui doivent être contenues dans le paramètre Data peut être calculé de la façon suivante: Index2 – Index1 + 1.

¹⁰³⁾ Seulement un objet par transmission pour de tels groupes.

7.4.3.1.2 Parameter values

Send(Non)ConfDataReq (RESPONDER):

Parameter	Req
Gtype	Copy of value from <i>UnsolicitedDataTransferMgmt(start)Ind</i> of the Primary FU (<i>Unsolicited Data Transfer FU</i>).
Gnr	Copy of value from <i>UnsolicitedDataTransferMgmt(start)Ind</i> of the Primary FU.
Index1 ¹⁰²⁾	Shall be > 0: Equal to the lowest Object number, in accordance with the CS(R), of the range of Object numbers whose data values are contained in the Data parameter (see below). Shall be equal to Index2, if Gtype = <i>Text-message-group</i> ¹⁰³⁾
Index2	Shall be >= value of Index1: Equal to the highest Object number, in accordance with the CS(R), of the range of Object numbers whose data values are contained in the Data parameter (see below). Shall be equal to Index1, if Gtype = <i>Text-message-group</i>
T	Time stamp, applying to the whole set of values contained in the Data parameter (below), and determined by the RESPONDER UE. The use of UTC is recommended.
Data	The actual data transferred. The structure is defined in annex A of IEC 60870-6-501.
Only for <i>SendNonConfDataReq</i> Result	= <i>result-ok</i>

SendConfDataResp (INITIATOR)

Parameter	Resp
Result	If no error detected by the INITIATOR UE: = <i>result-ok</i> . If error detected by the INITIATOR UE: Other value. See 7.4.3.2.

MixedDataReq (RESPONDER):

Parameter	Req
T	The point in time to which all relative time values coded into the Data parameter relate. The use of UTC is recommended.
Data	For each data value: – Group number: Value of attribute Group number. – IndexNumber: Value of attribute Object number – Delta Time: Time, relative to parameter T – Data: Quality code and value. The coding is described in annex A of IEC 60870-6-501.

¹⁰²⁾ The number of values that shall be contained in the Data parameter can be computed as: Index2 – Index1 + 1.

¹⁰³⁾ Only one object per transmission for such groups.

7.4.3.2 Traitement des erreurs

7.4.3.2.1 FU disruption

Lorsqu'elles sont interrompues, les deux parties de l'invocation en cours de la *Unsolicited Periodic Data Transfer FU* doivent se terminer correctement, aucune des deux parties ne devant tenter d'émettre une primitive associée à la fin de l'invocation elle-même¹⁰⁴⁾.

Une fois l'invocation terminée, la *Permanent Association FU* passe dans un état dans lequel elle doit signaler le code de redémarrage *Restart, spontaneous management lost* lorsqu'une autre association ayant les caractéristiques nécessaires à la *Unsolicited Periodic Data Transfer FU* est établie entre les mêmes membres de la paire INITIATOR/RESPONDER UE.

7.4.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Unsolicited Periodic Data Transfer FU* n'est pas présente dans un RESPONDER UE:

Les requêtes d'invocation sont toujours générées localement; voir 7.4.2.3. En conséquence, ce type d'erreur est traité localement et ne fait pas partie du domaine d'application du présent rapport technique.

FU présente, mais tentative illégale:

Les tentatives d'invocations illégales de la *Unsolicited Periodic Data Transfer FU* d'un groupe donné, dans un RESPONDER UE sont traitées localement et ne font pas partie du domaine d'application du présent rapport technique.

7.4.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>Send(Non)ConfData(spont.)Ind</i> <i>MixedDataInd</i>
FU non en fonctionnement	Ignorer, ou: Simuler l'entrée en phase 3 d'une invocation de <i>Unsolicited Periodic Data Transfer FU</i> pour le groupe considéré et terminer l'invocation en cours (supposée) de la <i>Unsolicited Periodic Data Transfer FU</i> dans le RESPONDER UE.
En attente de la primitive <i>Send(Non)ConfData(spont.)Ind</i>	(Normal)

Partie RESPONDER:

Etat	<i>SendConfDataConf</i>
FU non en fonctionnement	Ignorer ou indication d'erreur locale/connexion.
FU en fonctionnement, en attente d'une primitive <i>SendConfDataConf</i>	(Normal)

¹⁰⁴⁾ Les procédures de nettoyage locales ne sont pas spécifiées par le présent rapport technique.

7.4.3.2 Error handling

7.4.3.2.1 FU disruption

When disrupted, both parts of the current invocation of the *Unsolicited Periodic Data Transfer FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself¹⁰⁴).

Following the termination, the *Permanent Association FU* will enter a state in which it shall signal *Restart*, *spontaneous management lost*, the next time an association with the characteristics for the *Unsolicited Periodic Data Transfer FU* is established between the same INITIATOR and RESPONDER UE pair.

7.4.3.2.2 Illegal invocation attempt

FU not present:

If the *Unsolicited Periodic Data Transfer FU* is not present in a RESPONDER UE:

Invocation requests are always generated locally, see 7.4.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this technical report.

FU present, but attempt illegal:

Attempts of illegal invocations of the *Unsolicited Periodic Data Transfer FU* for any given group in a RESPONDER UE is a local issue, outside the scope of this technical report.

7.4.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>Send(Non)ConfData(spont)Ind</i> <i>MixedDataInd</i>
FU not running	Ignore or: Simulate the entry of Phase 3 of a <i>Unsolicited Periodic Data Transfer FU</i> invocation for the group in question, thus terminating the (supposed) current invocation of the <i>Unsolicited Periodic Data Transfer FU</i> in the RESPONDER UE.
Waiting for <i>Send(Non)ConfData (spont)Ind</i>	(Normal)

RESPONDER part:

State	<i>SendConfDataConf</i>
FU not running	Ignore or local error indication/logging.
FU running, waiting for <i>SendConfDataConf</i>	(Normal)

¹⁰⁴) Local clean-up procedures are not specified by this technical report.

7.4.3.2.4 Timing errors

Erreur dans la partie INITIATOR:

Erreur	Réaction de la part de TASE.1	Action spécifiée dans la FU
Emission trop tardive de la part de l'UE de la primitive <i>SendConfData(spont.)Resp</i> après la réception de la primitive <i>SendConfData(spont.)Ind</i> :	Dans l'INITIATOR: Erreur locale lors d'une éventuelle tentative d'émission de la primitive <i>SendConfData(spont.)Resp</i> Dans le RESPONDER: <i>SendConfData(spont.)Resp</i> avec Result = <i>remote-service-user-unavailable</i> .	Partie INITIATOR: Ignorer ou indication d'erreur locale/connexion, puis continuer normalement. Partie RESPONDER: Effectuer la procédure à mettre en place dans le cas d'un <i>Missing data acknowledgement</i> dans le RESPONDER. Voir 7.4.3.2.6

Erreur dans la partie RESPONDER:

Aucune.

7.4.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsque ce type d'erreur intervient au cours d'une tentative d'émission de *SendConfDataResp*:

- ignorer ou rapporter/ consigner l'erreur dans un journal (facultatif).

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsque ce type d'erreur intervient au cours d'une tentative d'émission de *Send(Non)ConfData Req* ou de *MixedData Req*:

- rapporter l'erreur localement, puis abandonner la tentative en réinitialisant le DATA TIMER à sa position normale.

Voir 4.2.1.

7.4.3.2.6 TASE.1 service primitive parameter errors

Le paramètre Gnr sera toujours valide dans le cadre de cette FU, à condition que la FU soit en fonctionnement¹⁰⁵⁾ pour ce groupe. Son rôle est d'identifier l'invocation de la FU vers laquelle la primitive entrante doit être dirigée.

¹⁰⁵⁾ Il convient que le cas de la non-invocation de la FU pour le groupe soit traité hors de la FU (par la Coordinating Function): La primitive de réponse naturelle doit être émise, avec le paramètre Result = *gnr-out-of-range*.

7.4.3.2.4 Timing errors

Error in INITIATOR part:

Error	Reaction from TASE.1	Specified action in FU
UE too late issuing <i>SendConfData(spont)Resp</i> after receiving <i>SendConfData(spont)Ind</i> :	In INITIATOR: Local error from eventual attempt at issuing <i>SendConfData (spont)Resp</i> In RESPONDER: <i>SendConfData(spont)Resp</i> , with Result = <i>remote-service-user-unavailable</i> .	INITIATOR part: Ignore, or local error indication/logging, then proceed as normal. RESPONDER part: Enter procedure for <i>Missing data acknowledgement</i> in RESPONDER. See 7.4.3.2.6.

Error in RESPONDER part:

None.

7.4.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with a *SendConfDataResp* attempt:

- ignore, or optionally report / log the error.

See 4.2.1.

RESPONDER part of the FU:

When occurring with a *Send(Non)ConfDataReq* or *MixedDataReq* attempt:

- report the error locally, then abandon the attempt, reinitializing the DATA TIMER as normal.

See 4.2.1.

7.4.3.2.6 TASE.1 service primitive parameter errors

The parameter Gnr will always be valid within the context of this FU, *provided the FU is running*¹⁰⁵⁾ for that group: it serves as identification of the FU invocation to which the incoming primitive shall be directed.

¹⁰⁵⁾ The case of the FU not being invoked for the group should be handled outside the FU (by the Coordinating Function): the natural replying primitive shall be issued, with parameter Result = *gnr-out-of-range*.

Erreurs dans la primitive *MixedDataInd* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
IndexNumber non valide (les conditions de validité sont définies en 7.4.3.1.2)	Ignorer les données et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).
Données non valides ou manquantes.	Terminer normalement l'association sur laquelle la <i>Unsolicited Periodic Data Transfer FU</i> est en fonctionnement.
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	Ignorer les données et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).
Security class 3: La somme de contrôle reçue >< somme de contrôle générée au cours du décryptage.	Ignorer les données et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).

Erreurs dans la primitive *SendConfData(spont.)Conf* (détectées par la partie RESPONDER):

Erreur	Action dans la partie INITIATOR de la FU
Result >< <i>result-ok</i>	Effectuer la procédure à mettre en place dans le cas d'une <i>Missing data confirmation</i> (voir ci-dessous)

Procédure à mettre en place dans le cas d'une Missing data confirmation:

- Si Result = *remote-service-user-unavailable* (partie INITIATOR non en cours de fonctionnement):
Terminer l'invocation de FU localement.
- Si Result = *misbehaviour-of-remote-service-user*:
Suivre les procédures normales.
- Si la valeur de Result est une valeur autre que *result-ok*, *spontaneous-transfer-not-initiated* ou *misbehaviour-of-remote-service-user*:
Réessayer. Le nombre de relances est défini localement et est compris entre 0 (pas de relance) et l'infini (relances illimitées).
 - Attendre pendant une durée déterminée localement.
 - Répéter toutes les primitives *SendConfData (spont.)Ind* non reconnues.
 Terminer la boucle de relances lorsque l'une des conditions (et actions) ci-dessus existe et que Result = *result-ok*, auquel cas:
Suivre la procédure normale.
Si tous les essais ont échoué (Result >< *result-ok*, pour tous les essais) ou qu'une décision locale interdit toute relance:
Suivre la procédure normale, ou
Déclencher la fin brutale de l'association.

Errors in the *MixedDataInd* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Invalid IndexNumber (Validity conditions are given above, in 7.4.3.1.2)	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
Invalid or missing Data	Perform a normal termination of the association over which the <i>Unsolicited Periodic Data Transfer FU</i> is running.
The security class 2 : the received authentication code >< the generated authentication code base on the received data.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
The security class 3 : the received checksum >< the generated checksum during the decipherment.	Ignore the data, but otherwise proceed as normal, optionally local error count.

Errors in the *SendConfData(spont)Conf* primitive (detected by the RESPONDER part):

Error	Action in RESPONDER part of FU
Result >< <i>result-ok</i>	Enter procedure for <i>Missing data confirmation</i> (immediately below).

Procedure for Missing data confirmation:

- If Result = *remote-service-user-unavailable* (INITIATOR part not running):

Terminate FU invocation locally.

- If Result = *misbehaviour-of-remote-service-user*

Proceed as normal.

- If Result any other value than *result-ok*, *spontaneous-transfer-not-initiated* and *misbehaviour-of-remote-service-user*:

Retry, a locally determined number of times, including 0 (no retry) and infinite (always retry).

- Wait, for a locally determined time span.
- Repeat all unacknowledged *SendConfData(spont)Ind*

Terminate retry loop on any of the conditions (and actions) above, as well as Result = *result-ok*, in which case:

Proceed as normal.

If all retrials failed (Result >< *result-ok*, for all retrials), or local decision not to retry at all:

Proceed as normal, or:

Trigger abrupt termination of supporting association

Erreurs dans la primitive *SendNonConfData(spont.)Ind* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Index1 ou Index2 non valide (les conditions de validité sont définies en 7.4.3.1.2)	Ignorer les données en incrémentant le compteur d'erreurs locales (facultatif).
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	Ignorer les données, et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).
Security class 3: La somme de contrôle reçue >< somme de contrôle générée au cours du décryptage.	Ignorer les données et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).
Données non valides ou manquantes	Terminer normalement l'association sur laquelle la <i>Unsolicited Periodic Data Transfer FU</i> est en cours de fonctionnement.

Erreurs dans la primitive *SendConfData(spont.)Ind* (détectées par la partie INITIATOR):

La partie INITIATOR doit dans tous les cas émettre une primitive *SendConfDataResp*, avec Result = <Valeur extraite du tableau ci-dessous>.

Erreur	Action dans la partie INITIATOR de la FU
Index1 ou Index2 non valide (les conditions de validité sont définies en 7.4.3.1.2)	<i>Index-out-of-range</i> .
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	<i>Authentication-failure</i> .
Security class 3: La somme de contrôle reçue >< somme de contrôle générée au cours du décryptage.	<i>Decipherment-error</i> .

7.5 Unsolicited Spontaneous Data Transfer FU

Type: Secondary.

7.5.1 Fonction

La *Unsolicited Spontaneous Data Transfer FU* est de type Secondary, ce qui signifie que l'invocation de la FU est déclenchée par le RESPONDER UE. La partie RESPONDER d'une *Unsolicited Spontaneous Data Transfer FU* est invoquée pour le data transfer. Cette invocation est effectuée en utilisant l'une des primitives de service suivantes:

- *SendConfData*;
- *SendNonConfData*;
- *MixedData*.

Le RESPONDER UE choisit le service à utiliser. Il est cependant permis à l'INITIATOR UE de proposer un service en utilisant le paramètre *PreferredService* lorsqu'il invoque la *Unsolicited Data Transfer FU*.

L'invocation de la *Unsolicited Spontaneous Data Transfer FU* suit la procédure suivante:

- a) La *Unsolicited Spontaneous Data Transfer FU* est invoquée pour le data transfer sur requête de la Coordinating Function locale, dans le RESPONDER UE. Les règles suivantes sont applicables:
 - Lorsqu'elle est invoquée, le moment d'invocations ainsi que la taille et le contenu des données sont déterminés par le RESPONDER UE.

Errors in the *SendNonConfData(spont)Ind* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Invalid Index1 or Index2 (Validity conditions are given above, in 7.4.3.1.2).	Ignore the data. Optionally incrementing local error count.
The security class 2: the received authentication code >< the generated authentication code based on the received data.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
The security class 3: the received checksum >< the generated checksum during the decipherment.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
Invalid or missing data.	Perform a normal termination on the association over which the <i>Unsolicited Periodic Data Transfer FU</i> is running.

Errors in the *SendConfData(spont)Ind* primitive (detected by the INITIATOR part):

The INITIATOR part shall in all these cases issue a *SendConfDataResp* primitive, with Result = <Value from table below>.

Error	Action in INITIATOR part of FU
Invalid Index1 or Index2 (Validity conditions are given above, in 7.4.3.1.2).	<i>Index-out-of-range.</i>
The security class 2: the received authentication code >< the generated authentication code based on the received data.	<i>Authentication-failure.</i>
The security class 3: the received checksum >< the generated checksum during the decipherment.	<i>Decipherment-error</i>

7.5 Unsolicited Spontaneous Data Transfer FU

Type: Secondary.

7.5.1 Function

The *Unsolicited Spontaneous Data Transfer FU* is of type Secondary, meaning that invocation of the FU is triggered by the RESPONDER UE. The RESPONDER part of an *Unsolicited Spontaneous Data Transfer FU* is invoked for data transfer. This is done by using one of the following service primitives:

- *SendConfData*,
- *SendNonConfData*,
- *MixedData*.

The decision of which service to use is taken by the RESPONDER UE. The INITIATOR UE, however, may propose a service by using the parameter *PreferredService* when invoking the *Unsolicited Data Transfer FU*.

Invocation of the *Unsolicited Spontaneous Data Transfer FU* applies with the following procedure:

- a) The *Unsolicited Spontaneous Data Transfer FU* is invoked for data transfer by request from the local Coordination Function in the RESPONDER UE. The following rules apply:
 - When invoked, the time of invocation, as well as the size and contents of the data part is determined by the RESPONDER UE.

- La *Unsolicited Data Transfer FU* doit être en cours de fonctionnement pour chaque groupe pour lequel la *Unsolicited Spontaneous Data Transfer FU* est à invoquer.
 - La séquence des valeurs de données de toute transmission est identique à celle de la définition du groupe.
 - Pour les groupes de type *Text-message-group*, chaque invocation de FU doit contenir la valeur de donnée d'un objet et d'un seul.
 - Une incarnation unique de la valeur de donnée de chaque objet doit toujours être contenue dans une transmission unique.
 - Il est permis au RESPONDER UE d'invoquer la *Unsolicited Spontaneous Data Transfer FU* autant de fois qu'il le souhaite, jusqu'à ce que l'INITIATOR UE retire l'autorisation d'émettre. Cette invocation est effectuée en terminant de façon ordonnée la *Unsolicited Data Transfer FU*.
 - Il est permis au RESPONDER UE d'utiliser le service de données confirmées, non confirmées ou mixtes spécifié sur la base d'un groupe. Le RESPONDER UE choisit le service à utiliser, sur proposition émise par le biais du paramètre *PreferredService* de la primitive de service *UnsolicitedDataTransferMgt(start,spont.)*.
 - Si le service *MixedData* est utilisé, les données sont associées à un groupe TASE.1 spécifié et sont soumises aux deux restrictions suivantes:
 - 1) L'intervalle de temps entre les valeurs ne doit pas être trop grand.
Voir CEI 60870-6-502, annexe A.
 - 2) La transmission de groupes de type *Text-message-group* n'est pas autorisée.
- b) L'INITIATOR UE reconnaît la réception des données chaque fois que le RESPONDER UE choisit une primitive de service *SendConfData(spont.)Req*

7.5.2 Règles de coordination

7.5.2.1 Utilisation d'une association

Toutes les interactions faisant partie d'une invocation de la *Unsolicited Spontaneous Data Transfer FU* sont véhiculées par une association unique. L'association doit être permanente et avoir les caractéristiques spécifiées en 7.5.2.3.1.

7.5.2.2 Relation avec les autres FU

7.5.2.2.1 Invoking FUs

La *Unsolicited Spontaneous Data Transfer FU* est invoquée à la demande de la Coordinating Function locale du RESPONDER UE.

7.5.2.2.2 Invoked FUs

La *Unsolicited Spontaneous Data Transfer FU* ne doit invoquer aucune autre FU.

7.5.2.2.3 Disrupting FUs

Il est permis aux FU suivantes d'interrompre la *Unsolicited Spontaneous Data Transfer FU*:

- *Permanent Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.5.2.2.4 Disrupted FUs

La *Unsolicited Spontaneous Data Transfer FU* ne doit interrompre aucune autre FU.

- The *Unsolicited Data Transfer FU* must currently be running for each group for which the *Unsolicited Spontaneous Data Transfer FU* is to be invoked.
 - The sequence of the data values in any transmission is the same as in the group definition.
 - For groups of type *Text-message-group*, each FU invocation shall contain the data value of exactly one object.
 - One single incarnation of the data value of any one object shall always be contained within a single transmission.
 - The RESPONDER UE may invoke the *Unsolicited Spontaneous Data Transfer FU* any number of times, until the INITIATOR UE withdraws the permission to send. This is done by orderly terminating the *Unsolicited Data Transfer FU*.
 - The RESPONDER UE may use the specified confirmed, non-confirmed or mixed data service on a per group basis. The RESPONDER UE decides which service to use, having received a suggestion by the *UnsolicitedDataTransferMgmt(start,spont.)* service primitive parameter *Preferred Service*.
 - If the *MixedData* service is used, the data is associated with one specified TASE.1 group, subject only to two limitations:
 - 1) The time span between values must not be too great; see IEC 60870-6-502, annex A.
 - 2) Transmitting groups of type *Text-message-group* is not allowed.
- b) The INITIATOR UE acknowledges the reception of data, whenever the RESPONDER UE has chosen the *SendConfData(spont)Req* service primitive.

7.5.2 Coordination rules

7.5.2.1 Association usage

All interactions that are part of one invocation of the *Unsolicited Spontaneous Data Transfer FU* are conveyed by one single association. The association shall be permanent, and have the characteristics as specified in 7.5.2.3.1.

7.5.2.2 Relation to other FUs

7.5.2.2.1 Invoking FUs

The *Unsolicited Spontaneous Data Transfer FU* is invoked by request from the local Coordinating Function in the RESPONDER UE.

7.5.2.2.2 Invoked FUs

The *Unsolicited Spontaneous Data Transfer FU* shall not invoke any other FU.

7.5.2.2.3 Disrupting FUs

The *Unsolicited Spontaneous Data Transfer FU* may be disrupted by:

- *Permanent Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.5.2.2.4 Disrupted FUs

The *Unsolicited Spontaneous Data Transfer FU* shall not disrupt any other FU.

7.5.2.3 Invocation

7.5.2.3.1 Conditions préalables

Les FU suivantes doivent avoir été invoquées avant toute invocation de la *Unsolicited Spontaneous Data Transfer FU*:

- *Permanent Association FU*;
- *Group Configuration FU*;
- Plusieurs invocations de *Unsolicited Data Transfer FU*.

L'invocation de la *Permanent Association FU* doit encore être en cours de fonctionnement au moment de l'invocation de la *Unsolicited Spontaneous Data Transfer FU*. L'association maintenue par l'invocation de la *Permanent Association FU* doit également être en cours de fonctionnement (et pas interrompue temporairement) au moment de l'invocation.

La *Permanent Association FU* doit avoir été invoquée pour pouvoir créer et maintenir l'association à utiliser pour les interactions liées à l'invocation en cours de la *Unsolicited Spontaneous Data Transfer FU*, en utilisant les caractéristiques suivantes:

- A-suffix (Application Entity Qualifier) tel qu'il est spécifié dans le tableau de 5.2.1.

La *Group Configuration FU* doit avoir été invoquée pour pouvoir définir et configurer le groupe à transmettre, avant l'invocation en cours de la *Unsolicited Periodic Data Transfer FU*.¹⁰⁶⁾

L'invocation de la *Group Configuration FU* de ce groupe ne doit pas être en fonctionnement au moment de l'invocation de la *Unsolicited Periodic Data Transfer FU*.

Les invocations de la *Unsolicited Data Transfer FU* doivent encore être en cours de fonctionnement au moment de l'invocation de la *Unsolicited Spontaneous Data Transfer FU* pour chaque groupe dont les données sont à transférer.

7.5.2.3.2 Restrictions

La *Unsolicited Spontaneous Data Transfer FU* ne doit pas être invoquée tant que l'une au moins des FU suivantes est en cours de fonctionnement, pour l'un au moins des groupes considérés:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.5.2.3.3 Invoking events

La partie INITIATOR de la *Unsolicited Spontaneous Data Transfer FU* est toujours invoquée par la réception de la première des trois primitives de service disponibles:

- *SendConfData(spont.)Ind*;
- *SendNonConfData(spont.)Ind*;
- *MixedDataInd*;

par le biais d'une association dont les caractéristiques sont définies pour la *Unsolicited Spontaneous Data Transfer FU*.

¹⁰⁶⁾ L'utilisation de la *Group Configuration FU* ou de la *Group Management FU* et de la *Group Definition FU* n'est cependant pas nécessaire si le groupe est défini dans la RESPONDER AP par un mécanisme non normalisé et est également déclaré dans la INITIATOR AP par un mécanisme non normalisé.

7.5.2.3 Invocation

7.5.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Unsolicited Spontaneous Data Transfer FU*:

- *Permanent Association FU*
- *Group Configuration FU*
- number of *Unsolicited Data Transfer FU* invocations.

The *Permanent Association FU* invocation shall still be running at the time of invocation of the *Unsolicited Spontaneous Data Transfer FU*, and the association maintained by the *Permanent Association FU* invocation shall be running (not temporarily broken) at that time.

The *Permanent Association FU* shall have been invoked to create and maintain the association to be used for the interactions related to the current invocation of the *Unsolicited Spontaneous Data Transfer FU*, with the following characteristics:

- A-suffix (Application Entity Qualifier) as specified in the table in 5.2.1.

The *Group Configuration FU* shall have been invoked to define and configure the group that is to be transmitted, prior to the current invocation of the *Unsolicited Periodic Data Transfer FU*.¹⁰⁶⁾

The *Group Configuration FU* invocation of that group shall not be running at the time of invocation of the *Unsolicited Periodic Data Transfer FU*.

The *Unsolicited Data Transfer FU* invocations shall still be running at the time of invocation of the *Unsolicited Spontaneous Data Transfer FU* for each group of which data is to be transferred.

7.5.2.3.2 Restrictions

The *Unsolicited Spontaneous Data Transfer FU* shall not be invoked while at least one of the following FUs are running, for at least one of the groups involved:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.5.2.3.3 Invoking events

The INITIATOR part of the *Unsolicited Spontaneous Data Transfer FU* is always invoked by reception of the first of the three available service primitives:

- *SendConfData(spont)Ind*
- *SendNonConfData(spont)Ind*
- *MixedDataInd*

via an association with the characteristics as defined for the *Unsolicited Spontaneous Data Transfer FU*.

¹⁰⁶⁾ The use of the *Group Configuration FU* or *Group Management FU* and *Group Definition FU* is, however, not required if the group is defined in the RESPONDER AP by a non-Standardized mechanism and also made known to the INITIATOR AP by a non-Standardized mechanism.

L'invocation de la partie RESPONDER de la *Unsolicited Spontaneous Data Transfer FU* est effectuée par un événement local après réception d'une primitive *UnsolicitedDataTransferMgmtInd(start,spont.)* valide.

7.5.2.4 Fin

7.5.2.4.1 Orderly termination

Il est permis aux seuls événements suivants de déclencher la fin ordonnée de la partie RESPONDER d'une invocation de *Unsolicited Spontaneous Data Transfer FU*:

- Requête locale par le biais de la Coordinating Function. L'origine de la requête ne fait pas partie du domaine d'application du présent rapport technique.
- Erreur d'encombrement. Voir 7.5.3.2.5.
- Réception d'une primitive *UnsolicitedDataTransferMgmtInd(stop,spont.)* valide.

7.5.2.4.2 Disruption

Il est permis aux événements suivants d'interrompre la partie INITIATOR et RESPONDER d'une invocation d'une *Unsolicited Spontaneous Data Transfer FU*:

- Interruption d'une autre invocation de FU. Voir 7.5.2.2.3.
- Condition d'erreur fatale rencontrée en fonctionnement. Voir 7.5.3.2.

7.5.3 Procédures

7.5.3.1 Primitives de service de TASE.1

Les services de TASE.1 élémentaires suivants sont utilisés par la *Unsolicited Spontaneous Data Transfer FU*:

- *SendConfData*;
- *SendNonConfData*;
- *MixedData*.

Le choix du service est effectué localement par le RESPONDER UE et ne fait pas partie du domaine d'application du présent rapport technique. L'INITIATOR UE doit faire une proposition de primitive de service à utiliser par le paramètre *PreferredService*, qui appartient à la primitive *UnsolicitedDataTransferMgmtInd(Start, Periodic,<PreferredService>)*, où *PreferredService* est susceptible de prendre l'une des trois valeurs suivantes:

- *SendConfData*,
- *SendNonConfData*,
- *SendMixedData*.

Il est permis au RESPONDER UE de choisir indépendamment entre les primitives de service disponibles chaque fois que la *Unsolicited Spontaneous Data Transfer FU* est invoquée.

7.5.3.1.1 Sequence

La séquence normale de primitives dépend du service choisi sur la Coordinating Function locale du RESPONDER UE:

Invocation of the RESPONDER part of the *Unsolicited Spontaneous Data Transfer FU* is done by a local event after a valid *UnsolicitedDataTransferMgmtInd(start,spont.)* primitive is received.

7.5.2.4 Termination

7.5.2.4.1 Orderly termination

Orderly termination of the RESPONDER part of an *Unsolicited Spontaneous Data Transfer FU* invocation may only be triggered by:

- Local request via the Coordinating Function. The original source of the request is outside the scope of this technical report.
- Congestion error. See 7.5.3.2.5.
- Reception of a valid *UnsolicitedDataTransferMgmtInd(stop,spont)* primitive.

7.5.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of an *Unsolicited Spontaneous Data Transfer FU* invocation may be disrupted by:

- Disruption of another FU invocation. See 7.5.2.2.3.
- A fatal error condition encountered during operation. See 7.5.3.2.

7.5.3 Procedures

7.5.3.1 TASE.1 service primitives

The following elementary TASE.1 service is used by the *Unsolicited Spontaneous Data Transfer FU*:

- *SendConfData*
- *SendNonConfData*
- *MixedData*.

The use of this service is a local decision of the RESPONDER UE, which is outside the scope of this technical report. The INITIATOR UE shall suggest which service primitive to use by the parameter *Preferred Service*, belonging to the *UnsolicitedDataTransferMgmtInd(Start,Periodic,<Preferred Service>)* primitive, where *Preferred Service* can take one of three values;

- *SendConfData*,
- *SendNonConfData*,
- *SendMixedData*.

The RESPONDER UE may choose independently between the available service primitives, each time the *Unsolicited Spontaneous Data Transfer FU* is invoked.

7.5.3.1.1 Sequence

The normal sequence of primitives is dependent on which service is chosen at the RESPONDER UE's local Coordinating Function:

Cas n° 1: Utilisation de la primitive *SendConfData*

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendConfData(spont.)Ind</i>	<-----	<i>SendConfData(spont.)Req</i>
<i>SendConfData(spont.)Resp</i>	-----> . The sequence is repeated until the FU is ended.	<i>SendConfData(spont.)Conf</i>

Cas n° 2: Utilisation de la primitive *SendNonConfData*

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendNonConfData(spont.)Ind</i>	<-----	<i>SendNonConfData(spont.)Req</i>
<i>SendNonConfData(spont.)Ind</i>	<-----	<i>SendNonConfData(spont.)Req</i>

Cas n° 3: Utilisation de la primitive *MixedData*

INITIATOR UE	TASE.1	RESPONDER UE
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>

Les règles suivantes sont applicables:

- a) Le RESPONDER UE détermine la date et l'heure ainsi que l'ordre des primitives *Send (Non)ConfDataReq* ou *MixedDataReq* des différentes invocations de la *Unsolicited Spontaneous Data Transfer FU*. Le RESPONDER UE doit ordonner la queue de données sortantes en fonction de l'attribut *Priority* du groupe auquel appartiennent les données:
 - Il est permis d'envoyer les données d'une valeur *Priority* donnée avant toutes les données en attente dont la valeur *Priority* est plus grande.
 - Il est permis d'envoyer les données dont la valeur *Priority* est identique par ordre d'occurrence.
 - Lorsque la valeur *Priority* est zéro, la fonction de priorité doit être interprétée comme désactivée pour ces données. Le contrôle de priorité est désactivé et les données concernées doivent être placées à la fin de la queue de données sortantes.¹⁰⁷⁾
- b) Les règles établies en 7.5.1 sont également applicables.

¹⁰⁷⁾ Voir 6.2.

Alternative 1: *SendConfData* primitive is used

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendConfData(spont)Ind</i>	<-----	<i>SendConfData(spont)Req</i>
<i>SendConfData(spont)Resp</i>	-----> .. The sequence is repeated until the FU is ended.	<i>SendConfData(spont)Conf</i>

Alternative 2: *SendNonConfData* primitive is used

INITIATOR UE	TASE.1	RESPONDER UE
<i>SendNonConfData(spont)Ind</i>	<-----	<i>SendNonConfData(spont)Req</i>
<i>SendNonConfData(spont)Ind</i>	<-----	<i>SendNonConfData(spont)Req</i>

Alternative 3: *MixedData* primitive is used

INITIATOR UE	TASE.1	RESPONDER UE
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>
<i>MixedDataInd</i>	<-----	<i>MixedDataReq</i>

The following rules apply:

- a) The RESPONDER UE is responsible for determining the timing and ordering of the *Send(Non)ConfDataReq* or the *MixedDataReq* primitives for the different *Unsolicited Spontaneous Data Transfer FU* invocations. The RESPONDER UE shall order its outgoing data queue according to the attribute *Priority* of the group to which the data belongs:
 - Data of a given *Priority* value may be sent prior to all pending data of greater *Priority* value.
 - Data of equal *Priority* value may be sent in order of occurrence.
 - *Priority* equal to zero shall be interpreted as "priority function off for these data", disabling priority check for the data concerned and always appending these at the end of the outgoing data queue¹⁰⁷⁾.
- b) The rules stated in 7.5.1 also apply here.

¹⁰⁷⁾ See 6.2.

7.5.3.1.2 Parameter values

Send(Non)ConfDataReq (RESPONDER):

Paramètre	Req.
Gtype	Copie de la valeur contenue dans la primitive <i>UnsolicitedDataTransferMgmt(start)Ind</i> de la FU <i>Primary (Unsolicited Data Transfer FU)</i> .
Gnr	Copie de la valeur contenue dans la primitive <i>UnsolicitedDataTransferMgmt(start)Ind</i> de la FU <i>Primary</i> .
Index1 ¹⁰⁸⁾	Doit être > 0: Egal à l'Object number le plus faible, selon le CS(R), de la plage d'Object numbers dont les valeurs de données sont contenues dans le paramètre Data (voir ci-dessous). Doit être égal à Index2, si Gtype = <i>Text-message-group</i> . ¹⁰⁹⁾
Index2	Doit être > = valeur de Index1: Egal à l'Object number le plus élevé, selon le CS(R), de la plage d'Object numbers dont les valeurs de données sont contenues dans le paramètre Data (voir ci-dessous). Doit être égal à Index1, si Gtype = <i>Text-message-group</i> .
T	Date et heure de l'ensemble du jeu de valeurs contenues dans le paramètre Data (voir ci-dessous) et déterminées par le RESPONDER UE. Il est conseillé d'utiliser l'heure UTC.
Data	Données réelles transférées. La structure est définie dans la CEI 60870-6-501, annexe A.
Result	= <i>result-ok</i>

SendConfData Resp (INITIATOR)

Paramètre	Resp
Result	Si aucune erreur n'est détectée par l'INITIATOR UE: = <i>result-ok</i> . Si une erreur est détectée par l'INITIATOR UE: Autre valeur. Voir 7.5.3.2.

MixedData:

Paramètre	Req (RESPONDER)
T	Date à laquelle toutes les valeurs de temps relatives codées dans le paramètre Data sont liées. Il est conseillé d'utiliser l'heure UTC.
MixedData	Pour chaque valeur de données: – Group number: Valeur de l'attribut Group number. – Index: Valeur de l'attribut Object number – DeltaTime: Heure, par rapport au paramètre T. – Code de qualité et valeur. Le codage est décrit dans l'annexe A de la CEI 60870-6-501.

¹⁰⁸⁾ Le nombre de valeurs qui doit être contenu dans le paramètre Data est susceptible d'être calculé comme suit:
Index2 – Index1 + 1.

¹⁰⁹⁾ Pour de tels groupes, un seul objet par transmission.

7.5.3.1.2 Parameter values

Send(Non)ConfDataReq (RESPONDER):

Parameter	Req
Gtype	Copy of value from <i>UnsolicitedDataTransferMgmt(start)Ind</i> of the Primary FU (<i>Unsolicited Data Transfer FU</i>).
Gnr	Copy of value from <i>UnsolicitedDataTransferMgmt(start)Ind</i> of the Primary FU.
Index1 ¹⁰⁸⁾	Shall be > 0: Equal to the lowest Object number, in accordance with the CS(R), of the range of Object numbers whose data values are contained in the Data parameter (below). Shall be equal to Index2, if Gtype = <i>Text-message-group</i> ¹⁰⁹⁾
Index2	Shall be >= value of Index1: Equal to the highest Object number, in accordance with the CS(R), of the range of Object numbers whose data values are contained in the Data parameter (below). Shall be equal to Index1, if Gtype = <i>Text-message-group</i> .
T	Time stamp, applying to the whole set of values contained in the Data parameter (below), and determined by the RESPONDER UE. The use of UTC is recommended.
Data	The actual data transferred. The structure is defined in annex A of IEC 60870-6-501.
Result	= <i>result-ok</i>

SendConfDataResp (INITIATOR):

Parameter	Resp
Result	If no error detected by the INITIATOR UE: = <i>result-ok</i> . If error detected by the INITIATOR UE: Other value. See 7.5.3.2.

MixedData:

Parameter	Req (RESPONDER)
T	The point in time to which all relative time values coded into the Data parameter relate. The use of UTC is recommended.
MixedData	For each data value: – Group number: Value of attribute Group number. – Index: Value of attribute Object number – DeltaTime: Time, relative to parameter T – Quality code and value. The coding is described in annex A of IEC 60870-6-501.

¹⁰⁸⁾ The number of values that shall be contained in the Data parameter can be computed as: Index2 – Index1 + 1.

¹⁰⁹⁾ Only one object per transmission for such groups.

7.5.3.2 Traitement des erreurs

7.5.3.2.1 FU disruption

Lorsqu'elles sont interrompues, les deux parties de l'invocation en cours de la *Unsolicited Spontaneous Data Transfer FU* doivent se terminer correctement, aucune des deux ne devant tenter d'émettre une primitive associée à la fin de l'invocation elle-même.¹¹⁰⁾

Une fois l'invocation terminée, la *Permanent Association FU* passe dans un état dans lequel elle doit signaler le code de redémarrage *Restart, spontaneous management lost* lorsqu'une autre association ayant les caractéristiques nécessaires à la *Unsolicited Spontaneous Data Transfer FU* est établie entre les mêmes membres de la paire INITIATOR/RESPONDER UE.

7.5.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Unsolicited Spontaneous Data Transfer FU* n'est pas présente dans un INITIATOR UE:

La présence de cette FU dépend de la *Unsolicited Data Transfer FU* primaire.

FU présente:

La réinvocation de la *Unsolicited Spontaneous Data Transfer FU* est traitée normalement et il convient d'utiliser les conditions précédentes de l'attribut.

7.5.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>Send(Non)ConfData(spont.)Ind</i> <i>MixedDataInd</i>
FU non en fonctionnement	Ignorer, ou Simuler l'entrée d'une invocation de la <i>Unsolicited Spontaneous Data Transfer FU</i> pour le groupe considéré et terminer ainsi l'invocation en cours (supposée) de la <i>Unsolicited Spontaneous Data Transfer FU</i> dans le RESPONDER UE.
En attente de la primitive <i>Send(Non)ConfData (spont.)Ind</i>	(Normal)

Partie RESPONDER:

Etat	<i>SendConfDataConf</i>
FU non en fonctionnement	Ignorer, ou indication d'erreur locale/connexion
FU en fonctionnement, en attente de la primitive <i>SendConfDataConf</i>	(Normal)

¹¹⁰⁾ Les procédures de nettoyage locales ne sont pas spécifiées par le présent rapport technique.

7.5.3.2 Error handling

7.5.3.2.1 FU disruption

When disrupted, both parts of the current invocation of the *Unsolicited Spontaneous Data Transfer FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself¹¹⁰⁾.

Following the termination, the *Permanent Association FU* will enter a state in which it shall signal *Restart, spontaneous management lost*, the next time an association with the characteristics for *Unsolicited Spontaneous Data Transfer FU* is established between the same INITIATOR and RESPONDER UE pair.

7.5.3.2.2 Illegal invocation attempt

FU not present:

If the *Unsolicited Spontaneous Data Transfer FU* is not present in an INITIATOR UE:

The presence of this FU is depending on the primary *Unsolicited Data Transfer FU*.

FU present:

Re-invoking the *Unsolicited Spontaneous Data Transfer FU* is treated as normal and earlier attribute conditions should still be used.

7.5.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>Send(Non)ConfData(spont)Ind</i> <i>MixedDataInd</i>
FU not running	Ignore, or: Simulate the entry of an <i>Unsolicited Spontaneous Data Transfer FU</i> invocation for the group in question, thus terminating the (supposed) current invocation of the <i>Unsolicited Spontaneous Data Transfer FU</i> in the RESPONDER UE.
Waiting for <i>Send(Non)ConfData(spont)Ind</i>	(Normal)

RESPONDER part:

State	<i>SendConfDataConf</i>
FU not running	Ignore or local error indication/logging.
FU running, waiting for <i>SendConfDataConf</i>	(Normal)

¹¹⁰⁾ Local clean-up procedures are not specified by this technical report.

7.5.3.2.4 Timing errors

Erreur dans la partie INITIATOR:

Erreur	Réaction de TASE.1	Action spécifiée dans la FU
Emission trop tardive de l'UE de la primitive <i>SendConfData(spont.)Resp</i> après la réception de la primitive <i>SendConfData(spont.)Ind</i> :	Dans l'INITIATOR: Erreur locale lors d'une éventuelle tentative d'émission de la primitive <i>SendConfData(spont.)Resp</i> Dans le RESPONDER: <i>SendConfData(spont.)Resp</i> avec Result = <i>remote-service-user-unavailable</i>	Partie INITIATOR: Ignorer ou indication d'erreur locale/connexion, puis continuer normalement. Partie RESPONDER: Suivre la procédure à mettre en place dans le cas d'un Missing data acknowledgement dans le RESPONDER. Voir 7.5.3.2.6.

Erreur dans la partie RESPONDER:

Aucune.

7.5.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsque ce type d'erreur intervient au cours d'une tentative d'émission de *SendConfDataResp*:

- ignorer ou rapporter/consigner l'erreur dans un journal (facultatif).

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsque ce type d'erreur intervient au cours d'une tentative d'émission de *Send(Non) ConfDataReq* ou *MixedDataReq*:

- rapporter l'erreur localement, puis abandonner la tentative en réinitialisant le DATA TIMER à sa position normale.

Voir 4.2.1.

7.5.3.2.6 TASE.1 service primitive parameter errors

Le paramètre Gnr sera toujours valide dans le cadre de cette FU, à condition que la FU soit en fonctionnement¹¹¹⁾ pour ce groupe. Il sert à identifier l'invocation de la FU vers laquelle la primitive entrante doit être dirigée.

¹¹¹⁾ Il convient que le cas de la non-invocation de la FU pour le groupe soit traité hors de la FU (par la Coordinating Function): il convient que la primitive de réponse naturelle soit émise, avec le paramètre Result = *gnr-out-of-range*.

7.5.3.2.4 Timing errors

Error in INITIATOR part:

Error	Reaction from TASE.1	Specified action in FU
UE too late issuing <i>SendConfData(spont)Resp</i> after receiving <i>SendConfData (spont)Ind</i> :	In INITIATOR: Local error from eventual attempt at issuing <i>SendConfData(spont)Resp</i> In RESPONDER: <i>SendConfData(spont)Resp</i> , with Result = <i>remote-service-user-</i> <i>unavailable</i> .	INITIATOR part: Ignore, or local error indication/logging, then proceed as normal. RESPONDER part: Enter procedure for Missing data acknowledgement in RESPONDER. See 7.5.3.2.6.

Error in RESPONDER part:

None.

7.5.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with a *SendConfDataResp* attempt:

Ignore, or optionally report/log the error.

See 4.2.1.

RESPONDER part of the FU:

When occurring with a *Send(Non)ConfDataReq* or *MixedDataReq* attempt:

Report the error locally, then abandon the attempt, reinitializing the DATA TIMER as normal.

See 4.2.1.

7.5.3.2.6 TASE.1 service primitive parameter errors

The parameter Gnr will always be valid within the context of this FU, *provided the FU is running*¹¹¹⁾ for that group: it serves as identification of the FU invocation to which the incoming primitive shall be directed.

¹¹¹⁾ The case of the FU not being invoked for the group shall be handled outside the FU (by the Coordinating Function): The natural replying primitive shall be issued, with parameter Result = *gnr-out-of-range*.

Erreurs dans la primitive *MixedDataInd* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
IndexNumber non valide (les conditions de validité sont définies en 7.5.3.1.2).	Ignorer les données et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).
MixedData non valides ou manquantes	Terminer normalement l'association sur laquelle la <i>Unsolicited Spontaneous Data Transfer FU</i> est en cours de fonctionnement.
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	Ignorer les données et poursuivre normalement en incrémentant le compteur d'erreurs locales (facultatif).
Security class 3: La somme de contrôle reçue >< somme de contrôle générée au cours du décryptage.	Ignorer les données et poursuivre normalement en incrémentant le compteur d'erreur locales (facultatif).

Erreurs dans la primitive *SendConfData(spont.)Conf* (détectées par la partie RESPONDER):

Erreur	Action dans la partie RESPONDER de la FU
Result >< <i>result-ok</i>	Effectuer la procédure à mettre en place dans le cas d'une Missing data confirmation (voir ci-dessous).

Procédure à mettre en place dans le cas d'une Missing data confirmation:

- Si Result = *remote-service-user-unavailable* (partie INITIATOR non en cours de fonctionnement):

Terminer l'invocation de FU localement.

- Si Result = *misbehaviour-of-remote-service-user*:

Suivre les procédures normales.

- Si la valeur de Result est une valeur autre que *result-ok*, *spontaneous-transfer-not-initiated* ou *misbehaviour-of-remote-service-user*.

Réessayer. Le nombre de relances est défini localement et est compris entre 0 (pas de relance) et l'infini (relances illimitées).

- Attendre pendant une durée déterminée localement.
- Répéter toutes les primitives *SendConfData(spont.)Ind* non reconnues.

Terminer la boucle de relances lorsque l'une des conditions (et actions) ci-dessus existe et que Result = *result-ok*, auquel cas:

Suivre la procédure normale.

Si tous les essais ont échoué (Result >< *result-ok*, pour tous les essais) ou qu'une décision locale interdit toute relance:

Suivre la procédure normale, ou

Déclencher la fin brutale de l'association.

Errors in the *MixedDataInd* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Invalid IndexNumber (Validity conditions are given in 7.5.3.1.2.)	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count
Invalid or missing mixedData	Perform a normal termination of the association over which the <i>Unsolicited Spontaneous Data Transfer FU</i> is running.
The security class 2: the received authentication code >< the generated authentication code based on the received data.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
The security class 3: the received checksum >< the generated checksum during the decipherment.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.

Errors in the *SendConfData(spont)Conf* primitive (detected by the RESPONDER part):

Error	Action in RESPONDER part of FU
Result >< <i>result-ok</i>	Enter procedure for Missing data confirmation (immediately below).

Procedure for Missing data confirmation:

- If Result = *remote-service-user-unavailable* (INITIATOR part not running):

Terminate FU invocation locally

- If Result = *misbehaviour-of-remote-service-user*

Proceed as normal.

- If Result any other value than *result-ok*, *spontaneous-transfer-not-initiated* and *misbehaviour-of-remote-service-user*.

Retry, a locally determined number of times, including 0 (no retry) and infinite (always retry).

- Wait, for a locally determined time span.
- Repeat all-unacknowledged *SendConfData(spont)Ind*

Terminate retry loop on any of the conditions (and actions) above, as well as Result = *result-ok*, in which case:

Proceed as normal.

If all retrials failed (Result >< *result-ok*, for all retrials), or local decision not to retry at all:

Proceed as normal, or:

Trigger abrupt termination of supporting association.

Erreurs dans la primitive *SendNonConfData(spont.)Ind* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Index1 ou Index2 non valide (les conditions de validité sont définies en 7.5.3.1.2)	Ignorer les données. Incrémenter le compteur d'erreurs locales (facultatif).
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	Ignorer les données et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).
Security class 3: La somme de contrôle reçue >< la somme de contrôle générée au cours du décryptage.	Ignorer les données et poursuivre normalement, en incrémentant le compteur d'erreurs locales (facultatif).
Données non valides ou manquantes	Terminer normalement l'association sur laquelle la <i>Unsolicited Spontaneous Data Transfer FU</i> est en cours de fonctionnement.

Erreurs dans la primitive *SendConfData(spont.)Ind* (détectées par la partie INITIATOR):

La partie INITIATOR doit dans tous les cas émettre une primitive *SendConfDataResp*, avec Result = <Valeur extraite du tableau ci-dessous>.

Erreur	Action dans la partie INITIATOR de la FU
Index1 ou Index2 non valide (les conditions de validité sont définies en 7.5.3.1.2)	<i>Index-out-of-range</i> .
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	<i>Authentication-failure</i> .
Security class 3: La somme de contrôle reçue >< somme de contrôle générée au cours du décryptage.	<i>Decipherment-error</i>

7.6 Supervisory Control Data Transfer FU

Type: Primary.

7.6.1 Fonction

L'invocation de la *Supervisory Control Data Transfer FU* donne à l'INITIATOR UE qui effectue l'invocation un certain contrôle fonctionnel sur le RESPONDER UE associé en lui passant des valeurs de consigne ou des commandes binaires convenues précédemment.

Les commandes binaires et les valeurs de consigne doivent être définies en tant que groupes TASE.1 du type approprié.

Les liens sémantiques réels (interprétation fonctionnelle) des commandes binaires ne sont pas spécifiés dans le présent rapport technique et font donc l'objet d'un accord spécifique entre les paires INITIATOR/RESPONDER UE impliquées.

Il est permis d'invoquer la FU pour une action en une phase ou deux phases. Une action en une phase contient une transaction TASE.1 unique, alors qu'une action en deux phases contient une paire de transactions consécutives.

Dans une action en une phase, le RESPONDER UE doit tenter d'exécuter la commande qui lui a été passée auparavant.

Errors in the *SendNonConfData(spont)Ind* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Invalid Index1 or Index2 (validity conditions are given above, in 7.5.3.1.2.)	Ignore the data. Optionally incrementing local error count.
The security class 2: the received authentication code >< the generated authentication code based on the received data.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
The security class 3: the received checksum >< the generated checksum during the decipherment.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
Invalid or missing Data.	Perform a normal termination of the association over which the <i>Unsolicited Spontaneous Data Transfer FU</i> is running.

Errors in the *SendConfData(spont)Ind* primitive (detected by the INITIATOR part):

The INITIATOR part shall in all these cases issue a *SendConfDataResp* primitive, with Result = <Value from table below>.

Error	Action in INITIATOR part of FU
Invalid Index1 or Index2 (validity conditions are given above, in 7.5.3.1.2.)	<i>Index-out-of-range.</i>
The security class 2: the received authentication code >< the generated authentication code based on the received data.	<i>Authentication-failure.</i>
The security class 3: the received checksum >< the generated checksum during the decipherment.	<i>Decipherment-error.</i>

7.6 Supervisory Control Data Transfer FU

Type: Primary.

7.6.1 Function

A *Supervisory Control Data Transfer FU* invocation gives the invoking INITIATOR UE a certain functional control of the associated RESPONDER UE, by way of passing setpoint values or previously agreed-upon binary commands to it.

Binary commands and setpoint values shall be defined as TASE.1 groups of the relevant types.

Actual semantics (functional interpretation) of binary commands is not specified by this technical report, and is therefore subject to separate agreement between the INITIATOR/RESPONDER UE pairs involved.

The FU may be invoked for either a one-phase or a two-phase action. A one-phase action comprises a single TASE.1 transaction, while a two-phase action comprises a pair of consecutive transactions.

In a one-phase action, the RESPONDER UE shall attempt to execute the command passed to it at once.

Dans une action en deux phases, la première transaction de TASE.1 demande au RESPONDER UE de se préparer à exécuter une commande passée. La seconde transaction donne ensuite l'instruction au RESPONDER UE soit d'exécuter la commande préparée, soit d'abandonner l'action en deux phases, c'est-à-dire d'annuler l'effet de la première transaction de l'action en deux phases en cours.

7.6.2 Règles de coordination

7.6.2.1 Utilisation d'une association

Toutes les interactions de TASE.1 faisant partie d'une invocation de la *Supervisory Control Data Transfer FU* sont véhiculées par une association unique. L'association doit avoir les caractéristiques spécifiées en 7.6.2.3.1.

7.6.2.2 Relation avec les autres FU

7.6.2.2.1 Invoking FUs

La *Supervisory Control Data Transfer FU* ne doit être invoquée par aucune autre FU.

7.6.2.2.2 Invoked FUs

La *Supervisory Control Data Transfer FU* ne doit invoquer aucune autre FU.

7.6.2.2.3 Disrupting FUs

Il est permis aux FU suivantes d'interrompre la *Supervisory Control Data Transfer FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.6.2.2.4 Disrupted FUs

La *Supervisory Control Data Transfer FU* ne doit interrompre aucune autre FU.

7.6.2.3 Invocation

7.6.2.3.1 Conditions préalables

Les FU suivantes doivent avoir été invoquées avant toute invocation de la *Supervisory Control Data Transfer FU*:

- *Permanent Association FU* ou *Dynamic Association FU*¹¹²⁾;
- *Group Configuration FU*¹¹³⁾.

Les invocations de la *Permanent Association FU* ou de la *Dynamic Association FU* doivent encore être actives au moment de l'invocation de la *Supervisory Control Data Transfer FU*. Lorsqu'une invocation de la *Permanent Association FU* existe, l'association maintenue par l'invocation doit être active (pas interrompue temporairement) au moment de l'invocation.

¹¹²⁾ Soumis à une décision locale dans l'INITIATOR UE.

¹¹³⁾ Il est également permis à la *Coordinating Function* d'invoquer directement la *Group Management FU* et la *Group Definition FU* invoquées elles-mêmes par la *Group Configuration FU*.

In a two-phase action, the first TASE.1 transaction instructs the RESPONDER UE to prepare for execution of a command passed to it. The second transaction then instructs the RESPONDER UE to either execute the command that has been prepared, or to abort the two-phase action, i.e. undo the effect of the first transaction of the current two-phase action.

7.6.2 Coordination rules

7.6.2.1 Association usage

All TASE.1 interactions that are part of one invocation of the *Supervisory Control Data Transfer FU* are conveyed by one single association. The association shall have the characteristics as specified in 7.6.2.3.1.

7.6.2.2 Relation to other FUs

7.6.2.2.1 Invoking FUs

The *Supervisory Control Data Transfer FU* shall not be invoked by any other FU.

7.6.2.2.2 Invoked FUs

The *Supervisory Control Data Transfer FU* shall not invoke any other FU.

7.6.2.2.3 Disrupting FUs

The *Supervisory Control Data Transfer FU* may be disrupted by:

- *Permanent Association FU*;
- *Dynamic Association FU*;
- *Group Management FU*;
- *Group Definition FU*;
- *Restart Reconfigure FU*.

7.6.2.2.4 Disrupted FUs

The *Supervisory Control Data Transfer FU* shall not disrupt any other FU.

7.6.2.3 Invocation

7.6.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Supervisory Control Data Transfer FU*:

- *Permanent Association FU* or *Dynamic Association FU*¹¹²⁾
- *Group Configuration FU*¹¹³⁾

The *Permanent Association FU* or *Dynamic Association FU* invocation shall still be active at the time of invocation of the *Supervisory Control Data Transfer FU*. In the case of *Permanent Association FU*, the association maintained by the invocation shall be active (not temporarily broken) at that time.

¹¹²⁾ Subject to local decision in the INITIATOR UE.

¹¹³⁾ Alternatively, the *Group Management FU* and *Group Definition FU*, which are invoked by the *Group Configuration FU*, may have been invoked directly by the Coordinating Function.

L'invocation de la *Group Configuration FU* doit se terminer au moment de l'invocation de la *Supervisory Control Data Transfer FU*.

La *Permanent Association FU* ou la *Dynamic Association FU* doivent avoir été invoquées pour pouvoir créer (et, pour la *Permanent Association FU*, également maintenir) l'association à utiliser pour les interactions liées à l'invocation en cours de la *Supervisory Control Data Transfer FU*, avec la paire A-suffix spécifiée par le tableau de 5.2.1.

La *Group Configuration FU* doit avoir été invoquée pour pouvoir définir et configurer le groupe qui définit la commande ou la valeur de consigne considérées, avant l'invocation en cours de la *Supervisory Control Data Transfer FU*¹¹⁴⁾.

7.6.2.3.2 Restrictions

Pour toute combinaison de systèmes INITIATOR/RESPONDER, les invocations multiples simultanées de la *Supervisory Control Data Transfer FU* pour un groupe donné ne sont pas autorisées.

La *Supervisory Control Data Transfer FU* ne doit pas être invoquée tant que l'une au moins des FU suivantes est active, pour le groupe impliqué:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.6.2.3.3 Invoking events

Il est permis à l'événement suivant d'invoquer la partie INITIATOR de la *Supervisory Control Data Transfer FU*:

- requête locale par le biais de la Coordinating Function, dont l'origine ne fait pas partie du domaine d'application du présent rapport technique.

L'invocation de la partie RESPONDER de la *Supervisory Control Data Transfer FU* est tentée chaque fois qu'une primitive *CommandTransferInd* valide avec le paramètre Com.type = CBXC ou = IXC est reçue par le biais d'une association ayant les caractéristiques définies pour la *Supervisory Control Data Transfer FU*. Voir 5.5.

Si le paramètre Com.type = CBXC, la partie RESPONDER doit être invoquée pour une action en deux phases. La partie RESPONDER doit être invoquée pour une action en une phase si le paramètre Com.type = IXC.

7.6.2.4 Fin

7.6.2.4.1 Orderly termination

Il est permis aux événements suivants de déclencher une fin ordonnée de la partie INITIATOR d'une *Supervisory Control Data Transfer FU*:

- Erreur d'encombrement. Voir 7.6.3.2.5.
- Réception d'une primitive de service *CommandTransferConf* non valide après l'émission d'une primitive de service *CommandTransferReq* (fin causée par une erreur).

¹¹⁴⁾ L'utilisation de la *Group Configuration FU* n'est cependant pas nécessaire si le groupe est défini dans la RESPONDER AP par un mécanisme non TASE.1 et est également déclaré dans l'INITIATOR AP par un mécanisme non TASE.1.

The *Group Configuration FU* invocation shall be terminated at the time of invocation of the *Supervisory Control Data Transfer FU*.

The *Permanent Association FU* or *Dynamic Association FU* shall have been invoked to create (and, for the *Permanent Association FU*, also to maintain) the association to be used for the interactions related to the current invocation of the *Supervisory Control Data Transfer FU*, with the A-suffix pair specified in 5.2.1.

The *Group Configuration FU* shall have been invoked to define and configure the group defining the command or setpoint in question, prior to the current invocation of the *Supervisory Control Data Transfer FU*.¹¹⁴⁾

7.6.2.3.2 Restrictions

For any given INITIATOR/RESPONDER system combination, multiple simultaneous invocations of the *Supervisory Control Data Transfer FU* for any given group are not allowed.

The *Supervisory Control Data Transfer FU* must not be invoked while at least one of the following FUs are active, for the group involved:

- *Group Configuration FU*;
- *Group Management FU*;
- *Group Definition FU*.

7.6.2.3.3 Invoking events

The INITIATOR part of the *Supervisory Control Data Transfer FU* may be invoked by:

- local request via the Coordinating Function, the original source of which is outside the scope of this technical report.

Invocation of the RESPONDER part of the *Supervisory Control Data Transfer FU* is attempted whenever a valid *CommandTransferInd* primitive with parameter *Com.type* = *CBXC* or = *IXC* is received via an association with the characteristics as defined for the *Supervisory Control Data Transfer FU*: see 5.5.

If parameter *Com.type* = *CBXC*, the RESPONDER part shall be invoked for a two-phase action. The RESPONDER part shall be invoked for a one-phase action if parameter *Com.type* = *IXC*.

7.6.2.4 Termination

7.6.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of a *Supervisory Control Data Transfer FU* invocation may be triggered by:

- Congestion error. See 7.6.3.2.5.
- Reception of an invalid *CommandTransferConf* service primitive after issuing a *CommandTransferReq* service primitive (termination on error).

¹¹⁴⁾ The use of the *Group Configuration FU* is, however, not required if the group is defined in the RESPONDER AP by a non-TASE.1 mechanism and also made known to the INITIATOR AP by a non-TASE.1 mechanism.

- Réception d'une primitive de service *CommandTransferConf* valide après l'émission de la dernière primitive de service *CommandTransferReq* de l'invocation de la *Supervisory Control Data Transfer FU* en cours (fin normale).

Dans le cas d'une action en une phase, la partie RESPONDER d'une invocation de la *Supervisory Control Data Transfer FU* se termine toujours d'elle-même (fin normale) après l'émission d'une primitive de service *CommandTransferResp*.

Dans une action en deux phases, la partie RESPONDER d'une invocation de la *Supervisory Control Data Transfer FU* se termine d'elle-même normalement, soit après l'émission de la seconde primitive de service *CommandTransferResp* (fin normale), soit après l'émission de la première primitive de service *CommandTransferResp* avec le paramètre Result >< result-ok (fin causée par une erreur).

7.6.2.4.2 Disruption

Il est admis que les parties INITIATOR et RESPONDER d'une invocation de la *Supervisory Control Data Transfer FU* soit interrompues par:

- Une autre invocation de FU. Voir 7.6.2.2.3.
- Une condition d'erreur fatale rencontrée en fonctionnement. Voir 7.6.3.2.

7.6.3 Procédures

7.6.3.1 Primitives de service de TASE.1

Le service de TASE.1 élémentaire suivant est utilisé par la *Supervisory Control Data Transfer FU*:

CommandTransfer

7.6.3.1.1 Sequence

A) Cas de l'invocation dans le cadre d'une action en une phase:

Dans le cadre d'une action en une phase, la séquence de primitives correspond à une transaction *CommandTransfer* unique:

INITIATOR UE	TASE.1	RESPONDER UE
<i>CommandTransferReq</i>	----->	<i>CommandTransferInd</i>
<i>CommandTransferConf</i>	<-----	<i>CommandTransferResp</i>

Règles:

- A la réception d'une primitive de service *CommandTransferInd* valide avec le paramètre Timemode = *argument-not-used*, le RESPONDER UE doit immédiatement essayer d'exécuter la commande intégrée à la primitive. Si l'exécution de la commande réussit¹¹⁵⁾, et uniquement dans ce cas, le paramètre Result de la primitive de service *CommandTransferResp* doit avoir la valeur *result-ok*.
- A la réception d'une primitive de service *CommandTransferInd* valide avec le paramètre Timemode = *latest-point-of-time-when-command-can-be-issued-at-the-RTU-side*, le RESPONDER UE doit enregistrer la commande et prévoir de l'exécuter le plus tôt possible,

¹¹⁵⁾ L'évaluation de la transmission de commande réussie sur le processus est soumise à une décision locale dans le RESPONDER UE.

- Reception of a valid *CommandTransferConf* service primitive after issuing the last *CommandTransferReq* service primitive of the current *Supervisory Control Data Transfer FU* invocation (normal termination).

The RESPONDER part of a *Supervisory Control Data Transfer FU* invocation for one-phase action always terminates itself (normal termination) after issuing a *CommandTransferResp* service primitive.

In a two-phase action, the RESPONDER part of a *Supervisory Control Data Transfer FU* invocation terminates itself normally, either after issuing the second *CommandTransferResp* service primitive (normal termination), or after issuing the first *CommandTransferResp* service primitive with parameter Result >< *result-ok* (termination on error).

7.6.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Supervisory Control Data Transfer FU* invocation may be disrupted by:

- Another FU invocation. See 7.6.2.2.3.
- A fatal error condition encountered during operation. See 7.6.3.2.

7.6.3 Procedures

7.6.3.1 TASE.1 service primitives

The following elementary TASE.1 service is used by the *Supervisory Control Data Transfer FU*:

CommandTransfer

7.6.3.1.1 Sequence

A) If invoked for one-phase action:

The one-phase action sequence of primitives is a single *CommandTransfer* transaction:

INITIATOR UE	TASE.1	RESPONDER UE
<i>CommandTransferReq</i>	----->	<i>CommandTransferInd</i>
<i>CommandTransferConf</i>	<-----	<i>CommandTransferResp</i>

Rules:

- Upon reception of a valid *CommandTransferInd* service primitive with parameter Timemode = *argument-not-used*, the RESPONDER UE shall immediately try to execute the command embedded in the primitive. Only in case of successful¹¹⁵⁾ command execution shall the Result parameter in the *CommandTransferResp* service primitive have the value *result-ok*.
- Upon reception of a valid *CommandTransferInd* service primitive with parameter Timemode = *latest-point-of-time-when-command-can-be-issued-at-the-RTU-side*, the RESPONDER UE shall register the command and schedule it for execution as soon as possible,

¹¹⁵⁾ Evaluation of the successful command transmission to the process is subject to local decision in the RESPONDER UE.

mais pas après la date indiquée par le paramètre T de la primitive¹¹⁶⁾. Le paramètre Result de la primitive de service *CommandTransferResp* doit avoir la valeur *result-ok* uniquement si l'enregistrement de la commande a réussi. Le paramètre Timemode = *latest-point-of-time-when-command-can-be-issued-at-the-RTU-side* n'est pas obligatoirement supporté.

- c) A la réception d'une primitive de service *CommandTransferInd* valide avec le paramètre Timemode = *point-of-time-when-command-shall-be-issued-at-the-RTU-side*, le RESPONDER UE doit enregistrer la commande et prévoir de l'exécuter le plus tôt possible après (mais pas avant) la date donnée par le paramètre T de la primitive. Le paramètre Result de la primitive de service *CommandTransferResp* doit avoir la valeur *result-ok* uniquement si la commande a été enregistrée correctement. Le paramètre Timemode = *point-of-time-when-command-can-be-issued-at-the-RTU-side* n'est pas obligatoirement supporté.
- d) Pour les groupes de type *Binary-command-group*, la commande transportée par les primitives *CommandTransfer* doit contenir un objet et un seul, c'est-à-dire que la valeur des paramètres Index1 et Index2 doit toujours être identique.

B) Cas de l'invocation dans le cadre d'une action en deux phases:

La séquence de primitives est composée d'une paire de transactions *CommandTransfer* consécutives (voir A, ci-dessus) implémentant un mécanisme de vérification avant exécution:

Au cours de la première transaction, l'INITIATOR UE demande au RESPONDER UE d'enregistrer la commande transportée par la primitive de service *CommandTransferInd* et de la préparer pour l'exécuter à un moment non défini, déterminé ultérieurement par l'INITIATOR UE. Le RESPONDER UE accepte la requête en renvoyant le paramètre Result = *result-ok* et accepte le rejet en renvoyant une autre valeur Result.

La seconde transaction doit uniquement être entrée (par l'INITIATOR UE) si la première transaction a renvoyé Result = *result-ok*. Au cours de la seconde transaction, il est permis à l'INITIATOR UE de donner l'instruction au RESPONDER UE:

- soit d'exécuter la commande enregistrée au cours de la première transaction, avec le paramètre Com.type = *EXC*;
- soit d'annuler l'action de la première transaction, en abandonnant l'action en deux phases, en utilisant le paramètre Com.type = *IHC* ou = *IXC*.

Les règles a) à d) de l'action en une phase s'appliquent également à la seconde transaction de l'action en deux phases.

¹¹⁶⁾ A noter que si la commande échoue, le RESPONDER UE n'est pas en mesure d'en informer immédiatement l'INITIATOR UE.

but not later than the point in time given by the parameter T of the primitive¹¹⁶⁾. The Result parameter in the *CommandTransferResp* service primitive shall have the value *result-ok* only if the command has been successfully registered. The support of Timemode = *latest-point-of-time-when-command-can-be-issued-at-the-RTU-side* is optional.

- c) Upon reception of a valid *CommandTransferInd* service primitive with parameter Timemode = *point-of-time-when-command-shall-be-issued-at-the-RTU-side*, the RESPONDER UE shall register the command and schedule it for execution as soon as possible after (but not before) the point in time given by the parameter T of the primitive. The Result parameter in the *CommandTransferResp* service primitive shall have the value *result-ok* only if the command has been successfully registered. The support of Timemode = *point-of-time-when-command-can-be-issued-at-the-RTU-side* is optional.
- d) For groups of type *Binary-command-group*, the command carried by *CommandTransfer* primitives shall contain exactly one object, i.e. the value of parameters Index1 and Index2 shall always be identical.

B) If invoked for two-phase action:

The primitive sequence is composed of a pair of consecutive *CommandTransfer* transactions (see A, above), implementing a "*check before execute*" mechanism:

During the first transaction, the INITIATOR UE effectively requests the RESPONDER UE to register the command carried by the *CommandTransferInd* service primitive and prepare it for execution at some indefinite time, to be decided upon later by the INITIATOR UE. The RESPONDER UE will accept the request by returning parameter Result = *result-ok*, and rejection by returning any other Result value.

The second transaction shall only be entered (by the INITIATOR UE) in case the first transaction returned Result = *result-ok*. During the second transaction, the INITIATOR UE may instruct the RESPONDER UE to either:

- execute the command registered during the first transaction, with parameter Com.type = *EXC*;

or:

- undo the action of the first transaction, effectively aborting the two-phase action, with parameter Com.type = *IHC* or = *IXC*.

Rules a) through d) for the one-phase action also apply for the second transaction of the two-phase action.

¹¹⁶⁾ Note that if the command eventually fails, the RESPONDER UE has no immediate way of reporting this back to the INITIATOR UE.

7.6.3.1.2 Parameter values

CommandTransfer:

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Gtype	One-phase action, or first transaction of two-phase action: <i>Valeur Binary-command-group, Analog-setpoint-group ou Digital-setpoint-group.</i> La valeur doit être égale à la valeur donnée de l'attribut Group type du CS(R) du groupe n° Gnr. Second transaction of two-phase action: Copie de la valeur à partir de la première transaction.	Copie de la valeur à partir de <i>Ind</i>
Gnr	One-phase action, or first transaction of two-phase action: Numéro de référence du groupe considéré. Valeur de l'attribut Group number d'un jeu d'attributs du CS(R) qui, avec son attribut Object set, définit la commande. Second transaction of two-phase action: Copie de la valeur à partir de la première transaction.	Copie de la valeur à partir de <i>Ind</i>
Index1	One-phase action, or first transaction of two-phase action: Borne inférieure d'une plage de données adjacentes de l'attribut Object number faisant partie de la définition de la commande. Doit être ≥ 1 et $\leq$ valeur donnée de l'attribut Group size du groupe, dans le CS(R). Si Gtype = <i>Binary-command-group</i> : Doit être = Index2. Second transaction of two-phase action: Copie de la valeur à partir de la première transaction.	Copie de la valeur à partir de <i>Ind</i>
Index2	One-phase action, or first transaction of two-phase action: Borne supérieure d'une plage de données adjacentes de l'attribut Object number faisant partie de la définition de la commande. Doit être $\geq$ Index1 et $\leq$ valeur donnée de l'attribut Group size du groupe, dans le CS(R). Si Gtype = <i>Binary-command-group</i> : Doit être = Index1. Second transaction of two-phase action: Copie de la valeur à partir de la première transaction.	Copie de la valeur à partir de <i>Ind</i>
T	One-phase action, or first transaction of two-phase action: Si Timemode = <i>argument-not-used</i> : Toutes les valeurs = -1. Si Timemode = <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> : La commande doit être exécutée avant ce moment. Si Timemode = <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> : la commande ne doit pas être exécutée avant ce moment. Second transaction of two-phase action: Copie de la valeur à partir de la première transaction.	Si Timemode = <i>argument-not-used</i> : Copie de la valeur à partir de <i>Ind</i> Si Timemode = <i>time-of-issuance</i> : Moment auquel la commande a été exécutée ou auquel il est prévu d'exécuter la commande.

7.6.3.1.2 Parameter values

CommandTransfer:

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Gtype	One-phase action, or first transaction of two-phase action: Value <i>Binary-command-group</i> , <i>Analog-setpoint-group</i> or <i>Digital-setpoint-group</i> . Value shall be equal to assumed value of attribute Group type in the CS(R) for group No. Gnr. Second transaction of two-phase action: Copy of value from first transaction.	Copy of value from <i>Ind</i>
Gnr	One-phase action, or first transaction of two-phase action: Reference number for the group in question. Value of attribute Group number in set of attributes in the CS(R), which together with its attribute Object set, defines the command. Second transaction of two-phase action: Copy of value from first transaction.	Copy of value from <i>Ind</i>
Index1	One-phase action, or first transaction of two-phase action: Lower bound of adjacent range of values of component attribute Object number taking part in defining the command. Shall be ≥ 1 and $\leq$ assumed value of attribute Group size for the group, in the CS(R). If Gtype = <i>Binary-command-group</i> : Shall be = Index2. Second transaction of two-phase action: Copy of value from first transaction.	Copy of value from <i>Ind</i>
Index2	One-phase action, or first transaction of two-phase action: Upper bound of adjacent range of values of component attribute Object number taking part in defining the command. Shall be $\geq$ Index1 and $\leq$ assumed value of attribute Group size for the group, in the CS(R). If Gtype = <i>Binary-command-group</i> : Shall be = Index1. Second transaction of two-phase action: Copy of value from first transaction.	Copy of value from <i>Ind</i>
T	One-phase action, or first transaction of two-phase action: If Timemode = <i>argument-not-used</i> : All values = -1. If Timemode = <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> : The time before which the command shall be executed. If Timemode = <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> : The time before which the command shall not be executed. Second transaction of two-phase action: Copy of value from first transaction.	If Timemode = <i>argument-not-used</i> : Copy of value from <i>Ind</i> If Timemode = <i>time-of-issuance</i> : The time at which the command has been executed, or at which the command is scheduled to be executed.

Paramètre	Req (INITIATOR)	Resp (RESPONDER)
Timemode	One-phase action: Doit être = <i>argument-not-used</i> . First transaction of two-phase action: Supportée obligatoirement: <i>argument-not-used</i> . Supportée en option: <i>latest-point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> et <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> . Second transaction of two-phase action: Copie de la valeur à partir de la première transaction	One-phase action: Doit être = <i>argument-not-used</i> . First transaction of two-phase action: Supportée obligatoirement: <i>argument-not-used</i> . Supportée en option: <i>latest-point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> et <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> . Second transaction of two-phase action: Copie de la valeur à partir de la première transaction.
Com.type	One-phase action: Doit être = <i>IXC</i> (exécution immédiate): Définit une action en une phase. First transaction of two-phase action: Doit être = <i>CBXC</i> (vérifier avant d'exécuter la commande): Définit une action en deux phases. Second transaction of two-phase action: Doit être = <i>EXC</i> (exécution de la commande) ou = <i>IHC</i> (interdiction d'exécuter la commande: annulation de la première phase). Si = <i>IXC</i> apparaît dans la seconde phase, il convient que l'action en deux phases soit terminée et qu'une action en une phase soit invoquée.	Si Com.type de <i>Ind</i> = <i>IXC</i> : Doit être = <i>EXR</i> . Si Com.type de <i>Ind</i> = <i>CBXC</i> : Doit être = <i>CBR</i> . Si Com.type de <i>Ind</i> = <i>EXC</i> : Doit être = <i>EXR</i> . Si Com.type de <i>Ind</i> = <i>IHC</i> : Doit être = <i>IHR</i> .
Data	Valeurs de l'objet de la commande réelle codées selon l'annexe A de la CEI 60870-6-501. Vide si Com.type = <i>CBXC</i> .	Copie de la valeur à partir de <i>Ind</i> Le code de qualité doit être fixé par le RESPONDER.
Result	(Non applicable)	Code fonctionnel ok/erreur Action effectuée comme spécifié: <i>result-ok</i> . Action non effectuée en raison d'une condition d'erreur: code d'erreur; voir 7.6.3.2.

7.6.3.2 Traitement des erreurs

7.6.3.2.1 FU disruption

Interruption par la *Permanent Association FU* ou la *Dynamic Association FU*:

L'interruption de la partie INITIATOR et RESPONDER de l'invocation en cours de la *Supervisory Control Data Transfer FU* doit être déclenchée localement. Elle doit être traitée en même temps que les primitives *TASEAbortInd* entrantes, dans la partie INITIATOR et dans la partie RESPONDER de l'invocation de la *Permanent Association FU* ou de la *Dynamic Association FU* traitant l'association sur laquelle la *Supervisory Control Data Transfer FU* est active.

Les deux parties de l'invocation en cours de la *Supervisory Control Data Transfer FU* doivent se terminer correctement, aucune des deux parties ne devant tenter d'émettre une primitive associée à la fin de l'invocation elle-même¹¹⁷⁾.

¹¹⁷⁾ Les procédures de nettoyage locales ne sont pas spécifiées par le présent rapport technique.

Parameter	Req (INITIATOR)	Resp (RESPONDER)
Timemode	One-phase action: Shall be = <i>argument-not-used</i> . First transaction of two-phase action: Mandatory supported: <i>argument-not-used</i> . Optionally supported: <i>latest-point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> and <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> . Second transaction of two-phase action: Copy of value from first transaction.	One-phase action: Shall be = <i>argument-not-used</i> . First transaction of two-phase action: Mandatory supported: <i>argument-not-used</i> . Optionally supported: <i>latest-point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> and <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> . Second transaction of two-phase action: Copy of value from first transaction.
Com.type	One-phase action: Shall be = <i>IXC</i> (immediate execute): Defines action as one-phase. First transaction of two-phase action: Shall be = <i>CBXC</i> (check before execute command): Defines action as two-phase. Second transaction of two-phase action: Shall be = <i>EXC</i> (execute command), or = <i>IHC</i> (inhibit command: Undo first phase). If = <i>IXC</i> appears in the second phase, the two-phase action should be terminated and a one-phase action invoked.	If Com.type in <i>Ind</i> = <i>IXC</i> : Shall be = <i>EXR</i> . If Com.type in <i>Ind</i> = <i>CBXC</i> : Shall be = <i>CBR</i> . If Com.type in <i>Ind</i> = <i>EXC</i> : Shall be = <i>EXR</i> . If Com.type in <i>Ind</i> = <i>IHC</i> : Shall be = <i>IHR</i> .
Data	Actual command object values, coded according to annex A of IEC 60870-6-501. May be empty, if Com.type = <i>CBXC</i> .	Copy of data value from <i>Ind</i> . The quality code shall be set by the RESPONDER.
Result	(Not applicable)	Functional OK/error code: Action performed as specified: <i>result-ok</i> . Action not performed, due to error condition: Error code; see 7.6.3.2.

7.6.3.2 Error handling

7.6.3.2.1 FU disruption

Disruption by the *Permanent Association FU* or *Dynamic Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Supervisory Control Data Transfer FU* shall be triggered locally, as a part of the handling of incoming *TASEAbortInd* primitives in both the INITIATOR part and the RESPONDER part of the *Permanent Association FU* or *Dynamic Association FU* invocation handling the association over which the *Supervisory Control Data Transfer FU* is active.

Both parts of the current invocation of the *Supervisory Control Data Transfer FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself¹¹⁷⁾.

¹¹⁷⁾ Local clean-up procedures are not specified by this technical report.

7.6.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Supervisory Control Data Transfer FU* n'est pas présente dans l'INITIATOR UE:

Ce type d'erreur, qui est généré par une tentative d'invocation par la Coordinating Function locale, est traité localement et ne fait pas partie du domaine d'application du présent rapport technique.

Si la *Supervisory Control Data Transfer FU* n'est pas présente dans un RESPONDER UE:

Le RESPONDER UE doit répondre aux tentatives d'activation par:

- émission d'une primitive *CommandTransferResp* avec Result = *remote-service-user-unavailable*.

FU présente mais tentative illégale:

Les tentatives d'invocation qui violent les règles établies dans l'INITIATOR UE, sont traitées localement et ne font pas partie du domaine d'application du présent rapport technique.

Dans un RESPONDER UE, les tentatives d'invocation qui violent ces mêmes règles doivent être traitées par le RESPONDER UE comme suit:

- émission d'une primitive *CommandTransferResp* avec Result = *remote-service-user-unavailable*, sans (ré)invocation de la *Supervisory Control Data Transfer FU* pour le groupe considéré, dans le RESPONDER UE.

7.6.3.2.3 Incoming TASE.1 service primitive out of context

Partie INITIATOR:

Etat	<i>CommandTransferConf</i>
FU non invoquée	Ignorer, ou indication d'erreur locale
Partie RESPONDER: Aucun	(Non applicable: Une primitive de service <i>CommandTransferInd</i> n'est jamais hors contexte dans le RESPONDER.)

7.6.3.2.4 Timing errors

Erreur dans la partie INITIATOR: Aucune.

Erreur dans la partie RESPONDER:

Erreur	Réaction de la part de TASE.1	Action spécifiée dans la FU
Retard de l'UE dans sa réponse à la primitive <i>CommandTransferInd</i>	Dans le RESPONDER: Erreur locale lors d'une éventuelle tentative d'émission de <i>CommandTransferResp</i> Dans l'INITIATOR: <i>CommandTransferConf</i>, avec Result = <i>remote-service-user-unavailable</i>.	Partie RESPONDER: Terminer l'invocation de la FU localement Partie INITIATOR: Terminer l'invocation de la FU localement.

7.6.3.2.2 Illegal invocation attempt

FU not present:

If the *Supervisory Control Data Transfer FU* is not present in an INITIATOR UE:

The handling of this type of error resulting from an invocation attempt by the local Coordinating Function is a local issue, outside the scope of this technical report.

If the *Supervisory Control Data Transfer FU* is not present in a RESPONDER UE:

The RESPONDER UE shall respond to activation attempts by:

- issuing a *CommandTransferResp* primitive with Result = *remote-service-user-unavailable*.

FU present, but attempt illegal:

Invocation attempts violating the rules stated in an INITIATOR UE is a local issue, outside the scope of this technical report.

In a RESPONDER UE, invocation attempts violating the same rules shall be handled by the RESPONDER UE in the following way:

- Issuing a *CommandTransferResp* primitive with Result = *remote-service-user-unavailable*, without (re-)invoking the *Supervisory Control Data Transfer FU* for the group concerned, in the RESPONDER UE.

7.6.3.2.3 Incoming TASE.1 service primitive out of context

INITIATOR part:

State	<i>CommandTransferConf</i>
FU not invoked	Ignore, or local error indication
RESPONDER part: None	(Not applicable: A <i>CommandTransferInd</i> service primitive is never out of context in the RESPONDER.)

7.6.3.2.4 Timing errors

Error in INITIATOR part: None.

Error in RESPONDER part:

Error	Reaction from TASE.1	Specified action in FU
UE too late responding to <i>CommandTransferInd</i>	In RESPONDER: Local error from eventual attempt at issuing <i>CommandTransferResp</i> In INITIATOR: <i>CommandTransferConf</i> , with Result = <i>remote-service-user-unavailable</i> .	RESPONDER part: Terminate FU invocation locally. INITIATOR part: Terminate FU invocation locally.

7.6.3.2.5 Congestion error

Partie INITIATOR de la FU:

Lorsque ce type d'erreur intervient avec une tentative d'émission de *CommandTransferReq*:

- terminer l'invocation de FU localement.

Voir 4.2.1.

Partie RESPONDER de la FU:

Lorsque ce type d'erreur intervient avec une tentative d'émission de *CommandTransferResp*:

- terminer l'invocation de FU localement.

7.6.3.2.6 TASE.1 service primitive parameter errors

Erreurs dans la primitive *CommandTransferInd* (détectées par la partie RESPONDER):

A) Action en une phase ou première transaction d'une action en deux phases:

La partie RESPONDER doit, dans tous ces cas, émettre une primitive *CommandTransferResp*, avec Result = <valeur extraite du tableau ci-dessous>, et les valeurs de Group-type, Group-number, Index1, Index2, T, Timemode, Com.type et Data identiques à celles de la primitive *Ind* correspondante.

La partie RESPONDER de la FU ne doit pas être invoquée.

Erreur	Valeur du paramètre Result
La valeur Group-type n'est pas égale à <i>Binary-command-group</i> , <i>Analog-setpoint-group</i> ou <i>Digital-setpoint-group</i> , ou n'est pas égale à la valeur de l'attribut Group type du CS(R) du groupe n° Gnr. (Le cas de l'absence de groupe n° Gnr est étudié ci-dessous.)	<i>gtype-out-of-range</i>
La valeur de Gnr est illégale, ou le groupe n° Gnr n'existe pas dans le CS(R)	<i>gnr-out-of-range</i>
Index1 <= 0 ou Index2 <= 0	<i>index-out-of-range</i>
Si Group-type = <i>Binary-command-group</i> : Index1 >< Index2	<i>index-out-of-range</i>
Si Group-type = <i>Analog-setpoint-group</i> ou = <i>Digital-setpoint-group</i> : Index1 est en dehors de la plage légale de valeurs de l'attribut Object number pour les objets définis dans le groupe n° Gnr du CS(R). Plage légale: 1 – Group size	<i>index-out-of-range</i>
Si Group-type = <i>Analog-setpoint-group</i> ou = <i>Digital-setpoint-group</i> : Index2 < Index1 ou Index2 > borne supérieure des valeurs de l'attribut Object number pour les objets définis dans le groupe n° Gnr du CS(R). Borne supérieure: Group size	<i>index-out-of-range</i>
Si Com.type >< <i>CBXC</i> ou (Com.type = <i>CBXC</i> et Data non vide): Nombre de valeurs de Data non égal à Index2 – Index1 + 1.	<i>index-out-of-range</i>
Timemode non égal à <i>argument-not-used</i> , <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> ou <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> .	<i>time-mode-out-of-range</i>
Timemode égal à <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> ou <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> , le RESPONDER UE ne supportant pas ce Timemode	<i>time-mode-not-supported-by-service-user</i>
Timemode = <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> ou <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> , et valeur de T illégale ou non égale à une date future.	<i>T-out-of-range</i>

7.6.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with a *CommandTransferReq* attempt:

- terminate FU invocation locally.

See 4.2.1.

RESPONDER part of the FU:

When occurring with a *CommandTransferResp* attempt:

- terminate FU invocation locally.

7.6.3.2.6 TASE.1 service primitive parameter errors

Errors in the *CommandTransferInd* primitive (detected by the RESPONDER part):

A) One-phase action, or first transaction of two-phase action:

The RESPONDER part shall in all these cases issue a *CommandTransferResp* primitive, with Result = <Value from table below>, and the values of Group-type, Group-number, Index1, Index2, T, Timemode, Com.type and Data as in the corresponding *Ind*.

The RESPONDER part of the FU shall not be invoked.

Error	Value of parameter Result
Group-type value not equal to <i>Binary-command-group</i> , <i>Analog-setpoint-group</i> or <i>Digital-setpoint-group</i> , or not equal to value of attribute Group type in CS(R) for group No. Gnr. (The case of non-existing group No. Gnr is considered below.)	<i>gtype-out-of-range</i>
Value of Gnr is illegal, or group No. Gnr does not exist in the CS(R)	<i>gnr-out-of-range</i>
Index1 <= 0 or Index2 <= 0	<i>index-out-of-range</i>
If Group-type = <i>Binary-command-group</i> : Index1 >> Index2	<i>index-out-of-range</i>
If Group-type = <i>Analog-setpoint-group</i> or = <i>Digital-setpoint-group</i> : Index1 outside legal range of values of component attribute Object number for defined objects in group No. Gnr in CS(R). Legal range: 1 – Group size.	<i>index-out-of-range</i>
If Group-type = <i>Analog-setpoint-group</i> or = <i>Digital-setpoint-group</i> : Index2 < Index1 or Index2 > upper bound of values of component attribute Object number for defined objects in group No. Gnr in CS(R). Upper bound: Group size.	<i>index-out-of-range</i>
If Com.type << <i>CBXC</i> , or (Com.type = <i>CBXC</i> and non-empty Data): Number of values in Data not equal to Index2 – Index1 + 1.	<i>index-out-of-range</i>
Timemode not equal to <i>argument-not-used</i> , <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> or <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> .	<i>time-mode-out-of-range</i>
Timemode equal to <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> or <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> , the RESPONDER UE not supporting this Timemode.	<i>time-mode-not-supported-by-service-user</i>
Timemode = <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> or <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> , and T value illegal or not a future point in time.	<i>T-out-of-range</i>

Erreur	Valeur du paramètre Result
Com.type différent des valeurs <i>IXC</i> , <i>CBXC</i> , <i>EXC</i> et <i>IHC</i>	<i>command-type-out-of-range</i>
Com.type = <i>IHC</i> ou <i>EXC</i>	<i>CBXC-not-received-before-EXC</i>
Illegal Data	<i>command-type-not-supported-by-service-user</i>
La commande spécifiée n'est pas en mesure d'être exécutée en raison de limites d'implémentation locales.	<i>command-type-not-supported-by-service-user</i>
Le RESPONDER UE n'est pas en mesure d'effectuer la fonction requise, pour toute raison autre que celles spécifiées ci-dessus.	<i>remote-service-user-unavailable</i>
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	<i>invalid-authentication-code-received</i>
Security class 3: La somme de contrôle reçue >< somme de contrôle générée au cours du décryptage.	<i>decipherment-error</i>

B) Seconde transaction d'une action en deux phases:

La partie RESPONDER doit, dans tous ces cas, émettre une primitive *CommandTransferResp*, avec Result = <Valeur extraite du tableau ci-dessous>, et les valeurs de Group-type, Group-number, Index1, Index2, T, Timemode, Com.type et Data identiques à celles de la primitive *Ind* correspondante.

La partie RESPONDER de la FU doit se terminer localement, sans tenter d'exécuter la commande enregistrée au cours de la précédente transaction.

Erreur	Valeur du paramètre Result
Group-type non égal au paramètre correspondant de la première transaction	<i>gtype-out-of-range</i>
Group-number non égal au paramètre correspondant de la première transaction	<i>gnr-out-of-range</i>
Index1 non égal au paramètre correspondant de la première transaction	<i>index-out-of-range</i>
Index2 non égal au paramètre correspondant de la première transaction	<i>index-out-of-range</i>
T non égal au paramètre correspondant de la première transaction	<i>T-out-of-range</i>
Timemode non égal au paramètre correspondant de la première transaction	<i>time-mode-out-of-range</i>
Timemode = <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> ou <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> et Com.type = <i>EXC</i> et valeur T non égale à une date future.	<i>T-out-of-range</i>
Com.type non égal à <i>EXC</i> ou <i>IHC</i> .	<i>command-type-out-of-range</i>
Data non égales au paramètre correspondant de la première transaction si data <> vides dans <i>CBXC</i> .	<i>EXC-data-different-from-CBXC-data</i>
Le RESPONDER UE n'est pas en mesure d'effectuer la fonction demandée pour une raison autre que celles énumérées ci-dessus.	<i>remote-service-user-unavailable</i>

Error	Value of parameter Result
Com.type different from values <i>IXC</i> , <i>CBXC</i> , <i>EXC</i> and <i>IHC</i>	<i>command-type-out-of-range</i>
Com.type = <i>IHC</i> or <i>EXC</i>	<i>CBXC-not-received-before-EXC</i>
Illegal Data	<i>command-type-not-supported-by-service-user</i>
Specified command cannot be executed, because of local implementation restrictions	<i>command-type-not-supported-by-service-user</i>
RESPONDER UE not able to perform the requested function, due to any reason other than those listed above.	<i>remote-service-user-unavailable</i>
For security class 2: The received authentication code $\neq$ the generated authentication code based in the received data.	<i>invalid-authentication-code-received</i>
For security class 3: The received checksum $\neq$ the generated checksum during the decipherment.	<i>decipherment-error</i>

B) Second transaction of two-phase action:

The RESPONDER part shall in all these cases issue an *CommandTransferResp* primitive, with Result = <Value from table below>, and the values of Group-type, Group-number, Index1, Index2, T, Timemode, Com.type, and Data as in the corresponding *Ind*.

The RESPONDER part of the FU shall be terminated locally, without attempting to execute the command registered during the first (previous) transaction.

Error	Value of parameter Result
Group-type not equal to corresponding parameter in first transaction	<i>gtype-out-of-range</i>
Group-number not equal to corresponding parameter in first transaction	<i>gnr-out-of-range</i>
Index1 not equal to corresponding parameter in first transaction	<i>index-out-of-range</i>
Index2 not equal to corresponding parameter in first transaction	<i>index-out-of-range</i>
T not equal to corresponding parameter in first transaction	<i>T-out-of-range</i>
Timemode not equal to corresponding parameter in first transaction	<i>time-mode-out-of-range</i>
Timemode = <i>latest-point-of-time-when-command-can-be-issued-at-the-RTU-side</i> or <i>point-of-time-when-command-shall-be-issued-at-the-RTU-side</i> , and Com.type = <i>EXC</i> and T value not a future point in time.	<i>T-out-of-range</i>
Com.type not equal to <i>EXC</i> or <i>IHC</i> .	<i>command-type-out-of-range</i>
Data not equal to corresponding parameter in first transaction if data $\neq$ empty in <i>CBXC</i> .	<i>EXC-data-different-from-CBXC-data</i>
RESPONDER UE not able to perform the requested function, due to any reason other than those listed above.	<i>remote-service-user-unavailable</i>

Erreurs dans la primitive *CommandTransferConf* (détectées par la partie INITIATOR):

Erreur	Action dans la partie INITIATOR de la FU
Absence de correspondance entre Gtype/Gnr/Index1/Index2/Data dans la primitive et Gtype/Gnr/Index1/Index2/Data dans la <i>CommandTransferReq</i> correspondante et Result = <i>result-ok</i> .	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Com.type dans la <i>Req</i> = <i>IXC</i> correspondante et Com.type dans la primitive non égales à <i>EXR</i> et Result = <i>result-ok</i> .	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Com.type dans la <i>Req</i> = <i>CBXC</i> correspondante et Com.type dans la primitive non égales à <i>CBR</i> et Result = <i>result-ok</i> .	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Com.type dans la <i>Req</i> = <i>EXC</i> correspondante et Com.type dans la primitive non égales à <i>EXR</i> et Result = <i>result-ok</i> .	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Com.type dans la <i>Req</i> = <i>IHC</i> correspondante et Com.type dans la primitive non égales à <i>IHR</i> et Result = <i>result-ok</i> .	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Result >< <i>result-ok</i>	Terminer l'invocation de la FU localement, en enregistrant l'erreur. Ne pas supposer que la commande a été acceptée ou exécutée.
Security class 2: Le code d'authentification reçu >< code d'authentification généré à partir des données reçues.	Terminer l'invocation de la FU localement, en enregistrant l'erreur.
Security class 3: La somme de contrôle reçue >< somme de contrôle générée au cours du décryptage.	Terminer l'invocation de la FU localement, en enregistrant l'erreur.

8 Function group redémarrage

8.1 Restart Reconfigure FU

Type: Composite.

8.1.1 Fonction

Une invocation de la *Restart Reconfigure FU* suit la procédure ci-dessous:

- L'INITIATOR UE interrompt les FU de transfert de données en cours d'un même système RESPONDER.
- L'INITIATOR UE invoque la *Dynamic Association FU* afin de créer une association et permettre ensuite à la *Group Configuration FU* de l'utiliser, si une telle association n'existe pas déjà.
- L'INITIATOR UE invoque la *Group Management FU* de façon à obtenir la fonction *Delete-all-groups* et à effacer la partie reconfigurable du CS(R).
- Pour chaque groupe du CS(I), l'INITIATOR UE invoque la *Group Configuration FU* de ce groupe en utilisant des données du CS(I), afin de rendre le CS(R) conforme au CS(I).
- L'INITIATOR UE invoque la *Restart Reactivate FU*, afin d'autoriser l'envoi de données attendues périodiquement ou non sollicitées par l'INITIATOR UE.
- L'INITIATOR UE termine finalement l'invocation de la *Dynamic Association FU* de manière ordonnée.
- L'INITIATOR UE traite les erreurs rencontrées entre les phases b) à e).

Errors in the *CommandTransferConf* primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Mismatch between Gtype/Gnr/Index1/Index2/Data in primitive and Gtype/Gnr/Index1/Index2/Data in corresponding <i>CommandTransfer Req</i> , and Result = <i>result-ok</i> .	Terminate the FU invocation locally, registering the error.
Com.type in corresponding <i>Req</i> = <i>IXC</i> , and Com.type in primitive not equal to <i>EXR</i> , and Result = <i>result-ok</i> .	Terminate the FU invocation locally, registering the error.
Com.type in corresponding <i>Req</i> = <i>CBXC</i> , and Com.type in primitive not equal to <i>CBR</i> , and Result = <i>result-ok</i> .	Terminate the FU invocation locally, registering the error.
Com.type in corresponding <i>Req</i> = <i>EXC</i> , and Com.type in primitive not equal to <i>EXR</i> , and Result = <i>result-ok</i> .	Terminate the FU invocation locally, registering the error.
Com.type in corresponding <i>Req</i> = <i>IHC</i> , and Com.type in primitive not equal to <i>IHR</i> , and Result = <i>result-ok</i> .	Terminate the FU invocation locally, registering the error.
Result $\neq$ <i>result-ok</i> .	Terminate the FU invocation locally, registering the error. Do not assume that the command has been accepted or executed.
For security class 2: The received authentication code $\neq$ the generated authentication code based in the received data.	Terminate the FU invocation locally, registering the error.
For security class 3: The received checksum $\neq$ the generated checksum during the decipherment.	Terminate the FU invocation locally, registering the error.

8 The Restart Function Group

8.1 Restart Reconfigure FU

Type: Composite.

8.1.1 Function

A *Restart Reconfigure FU* invocation adheres to the following procedure:

- The INITIATOR UE disrupts running data transfer FUs for the same RESPONDER system.
- The INITIATOR UE invokes the *Dynamic Association FU* in order to create an association for later use by the *Group Configuration FU*, if such association does not already exist.
- The INITIATOR UE invokes the *Group Management FU* for function *Delete-all-groups*, in order to clear the reconfigurable part of the CS(R).
- For each group in the CS(I), the INITIATOR UE invokes the *Group Configuration FU* for that group with data from the CS(I), in order to bring the CS(R) into accordance with the CS(I).
- The INITIATOR UE invokes the *Restart Reactivate FU*, in order to grant permissions to send the data that the INITIATOR UE expects to receive periodically or unsolicited.
- The INITIATOR UE eventually terminates the *Dynamic Association FU* invocation in an orderly manner.
- The INITIATOR UE processes errors encountered during phases b) through e).

8.1.2 Règles de coordination

8.1.2.1 Utilisation d'une association

Les interactions faisant partie de l'invocation de la *Group Management FU* et de la *Group Configuration FU* invoquées par la *Restart Reconfigure FU* sont véhiculées par une association ayant les caractéristiques spécifiées pour la *Group Management FU* et la *Group Configuration FU*.

8.1.2.2 Relation avec les autres FU

8.1.2.2.1 Invoking FUs

Il est permis aux FU suivantes d'invoquer la *Restart Reconfigure FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*.

8.1.2.2.2 Invoked FUs

Il est permis à la *Restart Reconfigure FU* d'invoquer les FU suivantes:

- *Dynamic Association FU*;
- *Group Management FU*;
- *Group Configuration FU*;
- *Restart Reactivate FU*.

8.1.2.2.3 Disrupting FUs

La *Restart Reconfigure FU* ne doit pas être interrompue. L'interruption des invocations de FU de support sont traitées comme des erreurs dans une invocation de la *Restart Reconfigure FU* mais elles n'interrompent pas cette dernière.

8.1.2.2.4 Disrupted FUs

Il est permis à la *Restart Reconfigure FU* de d'interrompre les FU suivantes:

- *Requested Data Transfer FU*;
- *Unsolicited Data Transfer FU*;
- *Unsolicited Periodic Data Transfer FU*;
- *Unsolicited Spontaneous Data Transfer FU*;
- *Supervisory Control Data Transfer FU*.

8.1.2.3 Invocation

8.1.2.3.1 Conditions préalables

Aucune invocation précédente ou concourante n'est requise.

8.1.2.3.2 Restrictions

Les invocations de la *Restart Reconfigure FU* ne sont pas autorisées dans une combinaison INITIATOR/RESPONDER UE donnée si:

- La *Restart Reconfigure FU* est déjà en fonctionnement pour la combinaison INITIATOR/RESPONDER UE ou

8.1.2 Coordination rules

8.1.2.1 Association usage

The interactions that are part of the *Group Management FU* and *Group Configuration FU* invocations that are invoked by the *Restart Reconfigure FU* are conveyed by an association with the characteristics as specified for the *Group Management FU* and the *Group Configuration FU*.

8.1.2.2 Relation to other FUs

8.1.2.2.1 Invoking FUs

The *Restart Reconfigure FU* may be invoked by the following FUs:

- *Permanent Association FU*;
- *Dynamic Association FU*.

8.1.2.2.2 Invoked FUs

The *Restart Reconfigure FU* may invoke the following FUs:

- *Dynamic Association FU*;
- *Group Management FU*;
- *Group Configuration FU*;
- *Restart Reactivate FU*.

8.1.2.2.3 Disrupting FUs

The *Restart Reconfigure FU* shall not be disrupted. Disruption of supporting FU invocations are treated as errors within a *Restart Reconfigure FU* invocation, but do not disrupt it.

8.1.2.2.4 Disrupted FUs

The *Restart Reconfigure FU* may disrupt:

- *Requested Data Transfer FU*;
- *Unsolicited Data Transfer FU*;
- *Unsolicited Periodic Data Transfer FU*;
- *Unsolicited Spontaneous Data Transfer FU*;
- *Supervisory Control Data Transfer FU*.

8.1.2.3 Invocation

8.1.2.3.1 Prerequisites

No previous or concurrent FU invocations are required.

8.1.2.3.2 Restrictions

For any given INITIATOR/RESPONDER UE combination, a *Restart Reconfigure FU* invocation is not allowed, if:

- *Restart Reconfigure FU* is already running for the INITIATOR/RESPONDER UE combination,

- La *Restart Reactivate FU* est déjà en fonctionnement pour la combinaison INITIATOR/RESPONDER UE¹¹⁸⁾.

8.1.2.3.3 Invoking events

En principe, il est admis d'invoquer la partie INITIATOR de la *Restart Reconfigure FU* chaque fois que l'INITIATOR UE l'estime approprié. Ce type d'invocation doit intervenir au moins après les événements suivants:

- a) Une erreur a été détectée dans la zone de contrôle de configuration¹¹⁹⁾ par la partie INITIATOR d'une invocation de *Permanent Association FU* ou de la *Dynamic Association FU*.
- b) Chaque fois que la *Restart Reconfigure FU* est à réinvoquer en raison de la présence d'erreurs dans les FU invoquées par la *Restart Reconfigure FU*, sur décision locale de l'INITIATOR UE.

La *Restart Reconfigure FU* est de type composite, ce qui signifie qu'il n'existe pas, dans la *Restart Reconfigure FU*, de partie RESPONDER spécifique ayant un événement invocateur spécifique associé.

8.1.2.4 Fin

8.1.2.4.1 Orderly termination

La fin ordonnée de la *Restart Reconfigure FU* n'est pas définie¹²⁰⁾. Toute fin ordonnée inattendue d'une FU invoquée par la *Restart Reconfigure FU* est considérée comme une erreur dans l'invocation de la *Restart Reconfigure FU*.

8.1.2.4.2 Disruption

Une invocation de la *Restart Reconfigure FU* ne doit pas être interrompue. Voir 8.1.2.2.3.

8.1.3 Procédures

8.1.3.1 Primitives de services de TASE.1

La *Restart Reconfigure FU* est de type composite, ce qui signifie qu'aucune séquence de primitives de service de TASE.1 ne lui est associée.

8.1.3.1.1 Séquence d'actions

La séquence d'actions normale se répartit en sept phases:

Phase 1: Interruption des invocations de FU en fonctionnement

Si une invocation de la *Data Transfer FU*¹²¹⁾ est en fonctionnement pour le même système RESPONDER, il faut qu'elle soit interrompue.

¹¹⁸⁾ En conséquence, tout événement déclenchant une invocation de la *Restart Reconfigure FU* ou de la *Restart Reactivate FU* intervenant dans le cadre de la *Restart Reconfigure FU* ou de la *Restart Reactivate FU* elle-même (déclenchée par une invocation de la *Dynamic Association FU* ou de la *Permanent Association FU*) sera soit ignorée, soit différée, jusqu'à ce que l'invocation de la *Restart Reconfigure FU* ou de la *Restart Reactivate FU* primaire soit terminée.

¹¹⁹⁾ Voir 5.4.

¹²⁰⁾ Ceci signifie qu'il faut que toute invocation de la *Restart Reconfigure FU* suive son cours et se termine d'elle-même.

¹²¹⁾ Pour obtenir une liste détaillée des FU appropriées, voir 8.1.2.2.4.

or:

- *Restart Reactivate FU* is already running for the INITIATOR/RESPONDER UE combination¹¹⁸⁾.

8.1.2.3.3 Invoking events

The INITIATOR part of the *Restart Reconfigure FU* may in principle be invoked whenever the INITIATOR UE determines that it is appropriate. At least, such invocation shall occur upon the following events:

- a) Configuration control field error¹¹⁹⁾ being detected by the INITIATOR part of a *Permanent Association FU* or *Dynamic Association FU* invocation.
- b) Whenever re-invocation of the *Restart Reconfigure FU* is to be made, because of errors in FUs invoked by the *Restart Reconfigure FU*, subject to local decision in the INITIATOR UE.

The *Restart Reconfigure FU* is of the composite type. Consequently, there exists no specific RESPONDER part of the *Restart Reconfigure FU*, with an associated specific invoking event.

8.1.2.4 Termination

8.1.2.4.1 Orderly termination

Orderly termination of the *Restart Reconfigure FU* is not defined¹²⁰⁾. Unexpected orderly termination of any FU that the *Restart Reconfigure FU* invokes is considered as an error within the *Restart Reconfigure FU* invocation.

8.1.2.4.2 Disruption

A *Restart Reconfigure FU* invocation shall not be disrupted. See 8.1.2.2.3.

8.1.3 Procedures

8.1.3.1 TASE.1 service primitives

The *Restart Reconfigure FU* is of the composite type, having no specific TASE.1 service primitive sequence associated with it.

8.1.3.1.1 Action sequence

The normal action sequence is partitioned into seven phases:

Phase 1: Disrupting of running FU invocations

If any *Data Transfer FU* ¹²¹⁾ invocation is running for the same RESPONDER system, it has to be disrupted.

¹¹⁸⁾ Consequently, any event triggering *Restart Reconfigure FU* or *Restart Reactivate FU* invocation occurring within the *Restart Reconfigure FU* or *Restart Reactivate FU* itself (triggered by a *Dynamic Association FU* or *Permanent Association FU* invocation) will have to be either ignored or deferred until the primary *Restart Reconfigure FU* or *Restart Reactivate FU* invocation is terminated.

¹¹⁹⁾ See 5.4.

¹²⁰⁾ This means that any *Restart Reconfigure FU* invocation will have to run its course, until it terminates itself.

¹²¹⁾ See 8.1.2.2.4 for a detailed list of relevant FUs.

Phase 2: Etablissement de l'association pour les invocations de la *Group Configuration FU*

Si aucune association ayant les caractéristiques requises n'existe déjà entre l'INITIATOR UE et le RESPONDER UE:

Si les conditions préalables et limites applicables à la *Dynamic Association FU* sont en mesure d'être respectées:

La *Dynamic Association FU* est invoquée par l'INITIATOR UE pour établir une association ayant les caractéristiques requises, entre l'INITIATOR UE et le RESPONDER UE.

Si la *Dynamic Association FU* fonctionne correctement:

L'invocation de la *Restart Reconfigure FU* passe à la phase 3.

Sinon:

L'invocation de la *Restart Reconfigure FU* traite¹²²⁾ l'erreur, puis passe à la phase 7.

Sinon:

L'invocation de la *Restart Reconfigure FU* traite l'erreur, puis passe à la phase 7.

Phase 3: Effacement de la partie reconfigurable du CS(R)

Si les conditions préalables et limites applicables à l'invocation d'une *Group Management FU* peuvent être respectées pour le groupe:

La *Group Management FU* est invoquée par l'INITIATOR UE pour la fonction *delete-all-groups*, afin d'effacer la partie du CS(R) qu'il est permis de modifier par le biais de la TAPI.

Si la *Group Management FU* fonctionne correctement:

La *Restart Reconfigure FU* passe à la phase 4.

Si la *Group Management FU* ne fonctionne PAS correctement ou ne se termine PAS normalement:

L'invocation de la *Restart Reconfigure FU* traite l'erreur, puis passe à la phase 7.

Sinon:

L'invocation de la *Restart Reconfigure FU* traite l'erreur, puis passe à la phase 7.

Phase 4: Rétablissement du CS(R) correct

Pour chaque groupe indiqué par le CS(I)¹²³⁾:

Si les conditions préalables et limites applicables à l'invocation de la *Group Configuration FU* du groupe sont en mesure d'être respectées:

La *Group Configuration FU* est invoquée par l'INITIATOR UE pour le groupe dont les données sont extraites du CS(I), afin de configurer le groupe dans le CS(R).

Si la *Group Configuration FU* fonctionne correctement:

La *Restart Reconfigure FU* passe à la phase 5.

Si la *Group Configuration FU* ne fonctionne PAS correctement ou ne se termine PAS normalement:

L'invocation de la *Restart Reconfigure FU* traite l'erreur, puis passe à la phase 7.

¹²²⁾ Le traitement des erreurs du Restart Function Group correspond généralement à un nombre de relances déterminé localement effectuées par l'INITIATOR UE.

¹²³⁾ C'est-à-dire uniquement les parties du CS jugées actuellement reconfigurables dans le CS(R) par le biais de l'interface Elcom. Il convient d'ignorer les groupes éventuels contenant l'attribut Persistent = true dans le CS(I).

Phase 2: Establishing the association for *Group Configuration FU* invocations

If an association with the desired characteristics does not already exist between the INITIATOR UE and the RESPONDER UE:

If the prerequisites and restrictions that apply to the *Dynamic Association FU* can be met:

The *Dynamic Association FU* is invoked by the INITIATOR UE, in order to establish an association with the desired characteristics between the INITIATOR UE and the RESPONDER UE.

If the *Dynamic Association FU* is running OK:

The *Restart Reconfigure FU* invocation enters phase 3.

If not:

The *Restart Reconfigure FU* invocation processes¹²²⁾ the error, then enters phase 7.

If not:

The *Restart Reconfigure FU* invocation processes the error, then enters phase 7.

Phase 3: Clearing the reconfigurable part of the CS(R)

If the prerequisites and restrictions that apply to a *Group Management FU* invocation for the group can be met:

The *Group Management FU* is invoked by the INITIATOR UE for function *delete-all-groups*, in order to clear the part of the CS(R) that may be modified via the TAPI.

If the *Group Management FU* is running ok:

The *Restart Reconfigure FU* enters phase 4.

If the *Group Management FU* is NOT running OK or NOT terminated normally:

The *Restart Reconfigure FU* invocation processes the error, then enters phase 7.

If not:

The *Restart Reconfigure FU* invocation processes the error, then enters phase 7.

Phase 4: Re-establishing correct CS(R)

For each group indicated by the CS(I)¹²³⁾:

If the prerequisites and restrictions that apply to a *Group Configuration FU* invocation for the group can be met:

The *Group Configuration FU* is invoked by the INITIATOR UE for the group with data from CS(I), in order to configure the group in the CS(R).

If the *Group Configuration FU* is running ok:

The *Restart Reconfigure FU* enters phase 5.

If the *Group Configuration FU* is NOT running OK or NOT terminated normally:

The *Restart Reconfigure FU* invocation processes the error, then enters phase 7.

If not:

The *Restart Reconfigure FU* invocation processes the error, then enters phase 7.

¹²²⁾ Within the Restart Function Group, processing of errors may generally comprise a locally determined number of retries on the part of the INITIATOR UE.

¹²³⁾ This is to be interpreted as only encompassing those parts of the CS that currently are supposed to be configurable in the CS(R) via the Elcom interface. Groups with attribute Persistent = *true* in the CS(I), if any, should be ignored.

Phase 5: Réactivation de transferts de données périodiques et spontanées

Si les conditions préalables et limites applicables à la *Restart Reactivate FU* sont en mesure d'être respectées:

La *Restart Reactivate FU* est invoquée par l'INITIATOR UE, afin d'autoriser le RESPONDER UE à émettre les données périodiques et spontanées attendues par le INITIATOR UE¹²⁴⁾.

Si la *Restart Reactivate FU* fonctionne correctement:

La *Restart Reconfigure FU* passe à la phase 6.

Si la *Restart Reactivate FU* ne fonctionne PAS correctement ou ne se termine PAS normalement:

L'invocation de la *Restart Reconfigure FU* traite l'erreur, puis passe à la phase 7.

Sinon:

L'invocation de la *Restart Reconfigure FU* traite l'erreur, puis passe à la phase 7.

Phase 6: Libération de l'association pour la Group Configuration FU

A un moment donné, déterminé localement par l'INITIATOR UE¹²⁵⁾, la *Dynamic Association FU* invoquée en phase 2 se termine de manière ordonnée. Les erreurs générées avec ce type de fin ne doivent avoir aucun effet sur les invocations de la *Restart Reconfigure FU*.

Phase 7: Traitement des erreurs

Si au moins une erreur sérieuse¹²⁶⁾ a été générée au cours de l'invocation de la *Restart Reconfigure FU* en cours, l'INITIATOR UE doit effectuer l'une des actions suivantes:

- a) Terminer l'invocation de la *Restart Reconfigure FU*, en signalant l'erreur pour réinitialiser le RESPONDER UE¹²⁷⁾.
- b) Terminer l'invocation de la *Restart Reconfigure FU*. Attendre pendant un moment déterminé localement, puis recommencer l'invocation de la *Restart Reconfigure FU* à partir du début.

Effectuer le cycle d'attente/réinvocation jusqu'à ce que l'invocation réussisse ou que le nombre d'invocations maximal déterminé localement soit atteint.

Si l'invocation n'est toujours pas réussie:

- terminer et signaler l'erreur, tel que décrit au point a), ci-dessus.

8.1.3.1.2 Parameter values

La FU est de type composite. Voir la description des FU en 4.2.2.

¹²⁴⁾ Par exemple, telles qu'elles sont déterminées dans la base de données d'état de l'INITIATOR UE.

¹²⁵⁾ La fin de la *Dynamic Association FU* ne doit pas être déclenchée avant que la phase 3 soit terminée. Par exemple, il est permis à l'INITIATOR UE de terminer la *Dynamic Association FU* immédiatement après cette phase ou après un délai. Dans ce dernier cas, il convient qu'un mécanisme "release-on-demand" (pour les ressources de TASE.1 occupées par l'invocation de la *Dynamic Association FU*) soit implémenté, autorisant la fin de l'invocation de la *Dynamic Association FU* avant l'expiration du délai.

¹²⁶⁾ Sur décision locale du INITIATOR UE.

¹²⁷⁾ La suite du traitement de l'échec de la réinitialisation se fait au niveau local, dans l'INITIATOR UE, et ne fait pas partie du domaine d'application du présent rapport technique. Cependant, ce type d'échec signifie qu'il existe une absence de correspondance persistante entre le CS(I) et le CS(R). Il convient donc de déclencher des actions de correction, généralement manuelles.

Phase 5: Re-activating periodic and spontaneous data transfers

If the prerequisites and restrictions that apply to the *Restart Reactivate FU* can be met:

The *Restart Reactivate FU* is invoked by the INITIATOR UE, in order to grant to the RESPONDER UE permission to send those periodic and spontaneous data that the INITIATOR UE expects¹²⁴⁾.

If the *Restart Reactivate FU* is running OK:

The *Restart Reconfigure FU* enters phase 6.

If the *Restart Reactivate FU* is NOT running OK or NOT terminated normally:

The *Restart Reconfigure FU* invocation processes the error, then enters phase 7.

If not:

The *Restart Reconfigure FU* invocation processes the error, then enters phase 7.

Phase 6: Releasing the association for the *Group Configuration FU*

At a point in time subject to local decision in the INITIATOR UE¹²⁵⁾, the *Dynamic Association FU* invoked in phase 2 is orderly terminated. Errors occurring with this termination shall have no effect on any *Restart Reconfigure FU* invocation.

Phase 7: Error handling.

If at least one serious¹²⁶⁾ error has been encountered during the current *Restart Reconfigure FU* invocation, the INITIATOR UE shall follow one of two courses of action:

- a) Terminate the *Restart Reconfigure FU* invocation, flagging failure to re-initialize the RESPONDER UE¹²⁷⁾.
- b) Terminate the *Restart Reconfigure FU* invocation. Wait for a locally determined amount of time, then re-invoke the *Restart Reconfigure FU* from the beginning.

Perform the wait/re-invoke cycle until success or a locally determined maximum number of invocations have been performed.

If still not successful:

Terminate, and flag the error, as described in a), above.

8.1.3.1.2 Parameter values

The FU is of the composite type. See description of component FUs in 4.2.2.

¹²⁴⁾ For example, as determined from a group state database in the INITIATOR UE.

¹²⁵⁾ *Dynamic Association FU* termination shall not be triggered before phase 3 is completed. The INITIATOR UE may for example terminate the *Dynamic Association FU* after completion of that phase, immediately or after a time delay. In the latter case, a "release-on-demand" mechanism for the TASE.1 resources occupied by the *Dynamic Association FU* invocation should be implemented, allowing termination the *Dynamic Association FU* invocation ahead of delay time expiry.

¹²⁶⁾ Subject to local decision in the INITIATOR UE.

¹²⁷⁾ Further processing of re-initialization failure is a local issue within the INITIATOR UE, outside the scope of this technical report. However, the failure means that there is a persistent mismatch between the CS(I) and the CS(R), so some sort of corrective action should eventually be triggered. Such corrective action will typically be manual.

8.1.3.2 Traitement des erreurs

Les erreurs générées avec les primitives de service de TASE.1 ne sont pas étudiées ici, car la *Restart Reconfigure FU* est de type composite.

Le traitement des erreurs générées dans les FU invoquées par la *Restart Reconfigure FU* est expliqué en 8.1.3.1.1.

8.1.3.2.1 FU disruption

La *Restart Reconfigure FU* ne peut pas être interrompue.

8.1.3.2.2 Illegal invocation attempt

FU non présente:

Si la *Restart Reconfigure FU* n'est pas présente dans un INITIATOR UE:

Les requêtes d'invocation sont toujours générées localement; voir 8.1.2.8. En conséquence, ce type d'erreur est traité localement et ne fait pas partie du domaine d'application du présent rapport technique.

Etant donné que la *Restart Reconfigure FU* est de type composite, aucune partie RESPONDER spécifique n'est définie pour cette FU.

FU présente, mais tentative illégale:

Les tentatives d'invocations multiples simultanées de la *Restart Reconfigure FU* d'une combinaison INITIATOR/RESPONDER UE donnée sont traitées localement et ne font pas partie du domaine d'application du présent rapport technique. L'INITIATOR UE doit cependant s'assurer que les échecs d'invocation de la *Restart Reconfigure FU* sont correctement enregistrés. Ces échecs signifient en effet qu'il existe une incohérence entre le CS(I) et le CS(R).

Il en va de même pour les tentatives d'invocation de la *Restart Reconfigure FU*, pour une combinaison INITIATOR/RESPONDER UE pour laquelle la *Restart Reactivate FU* est déjà en fonctionnement.

8.2 Restart Reactivate FU

Type: Composite.

8.2.1 Fonction

Une invocation de la *Restart Reactivate FU* suit la procédure ci-dessous:

- a) Pour chaque groupe du CS(I) que l'INITIATOR UE estime approprié, l'INITIATOR UE invoque la *Unsolicited Data Transfer FU* afin d'autoriser le RESPONDER UE à émettre les données appartenant au groupe, spontanément ou périodiquement.
- b) L'INITIATOR UE traite les erreurs rencontrées au cours de la phase a).

8.2.2 Règles de coordination

8.2.2.1 Utilisation d'une association

Les interactions de TASE.1 faisant partie des invocations de la *Unsolicited Data Transfer FU* invoquées par la *Restart Reactivate FU* sont véhiculées par les associations dont les caractéristiques sont spécifiées pour la *Unsolicited Data Transfer FU*.

8.1.3.2 Error handling

Errors occurring with TASE.1 service primitives are not considered here, since the *Restart Reconfigure FU* is of the composite type.

Handling of errors occurring in FUs invoked by the *Restart Reconfigure FU* is described in 8.1.3.1.1.

8.1.3.2.1 FU disruption

The *Restart Reconfigure FU* cannot be disrupted.

8.1.3.2.2 Illegal invocation attempt

FU not present:

If the *Restart Reconfigure FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see 8.1.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this technical report.

No specific RESPONDER part is defined for the *Restart Reconfigure FU*, since it is of the composite type.

FU present, but attempt illegal:

Attempts of multiple simultaneous invocations of the *Restart Reconfigure FU* for a given INITIATOR/RESPONDER UE combination is a local issue, outside the scope of this technical report. However, the INITIATOR UE must ensure that *Restart Reconfigure FU* failures are properly registered, since such failure signifies inconsistency between the CS(I) and the CS(R).

The same applies for attempts of invoking the *Restart Reconfigure FU* for an INITIATOR/RESPONDER UE combination for which the *Restart Reactivate FU* is already running.

8.2 Restart Reactivate FU

Type: Composite.

8.2.1 Function

A *Restart Reactivate FU* invocation adheres to the following procedure:

- a) For each group in the CS(I) that the INITIATOR UE determines appropriate, the INITIATOR UE invokes the *Unsolicited Data Transfer FU* in order to grant the RESPONDER UE permission to send data belonging to the group, spontaneously or periodically.
- b) The INITIATOR UE processes errors encountered during phase a).

8.2.2 Coordination rules

8.2.2.1 Association usage

The TASE.1 interactions that are part of the *Unsolicited Data Transfer FU* invocations invoked by the *Restart Reactivate FU* are conveyed by associations with the characteristics as specified for the *Unsolicited Data Transfer FU*.

8.2.2.2 Relation avec les autres FU

8.2.2.2.1 Invoking FUs

Il est permis aux FU suivantes d'invoquer la *Restart Reactivate FU*:

- *Permanent Association FU*;
- *Dynamic Association FU*;
- *Restart Reconfigure FU*.

8.2.2.2.2 Invoked FUs

Il est permis à la *Restart Reactivate FU* d'invoquer la FU suivante:

- *Unsolicited Data Transfer FU*.

8.2.2.2.3 Disrupting FUs

La *Restart Reactivate FU* ne doit pas être interrompue. Les interruptions d'invocations de FU de support sont traitées comme des erreurs dans une invocation de la *Restart Reactivate FU* mais elles n'interrompent pas cette dernière.

8.2.2.2.4 Disrupted FUs

La *Restart Reactivate FU* ne doit interrompre aucune autre FU.

8.2.2.3 Invocation

8.2.2.3.1 Conditions préalables

Aucune FU précédente ou concourante n'est nécessaire¹²⁸⁾.

8.2.2.3.2 Restrictions

Les invocations de la *Restart Reactivate FU* ne sont pas autorisées dans une combinaison INITIATOR/RESPONDER UE donnée si:

- la *Restart Reactivate FU* est déjà en fonctionnement pour la combinaison INITIATOR/RESPONDER UE, ou
- la *Restart Reconfigure FU* est déjà en fonctionnement pour la combinaison INITIATOR/RESPONDER UE.¹²⁹⁾

8.2.2.3.3 Invoking events

En principe, il est admis d'invoquer la partie INITIATOR de la *Restart Reactivate FU* chaque fois que l'INITIATOR UE l'estime approprié. Ce type d'invocation doit intervenir au moins après les événements suivants:

- a) Code de redémarrage *Restart*, spontaneous management lost reçu par la partie INITIATOR d'une invocation de la *Permanent Association FU* ou de la *Dynamic Association FU*.

¹²⁸⁾ L'invocation de la *Permanent Association FU* est requise par la *Unsolicited Data Transfer FU* invoquée par la *Restart Reactivate FU* et non par la *Restart Reactivate FU* elle-même.

¹²⁹⁾ En conséquence, tout événement déclenchant une invocation de la *Restart Reactivate FU* ou de la *Restart Reconfigure FU* intervenant dans le cadre de la *Restart Reactivate FU* ou de la *Restart Reconfigure FU* elle-même (déclenchée par une invocation de la *Dynamic Association FU* ou de la *Permanent Association FU*) sera soit à ignorer, soit à différer, jusqu'à ce que l'invocation de la *Restart Reactivate FU* ou de la *Restart Reconfigure FU* primaire soit terminée.

8.2.2.2 Relation to other FUs

8.2.2.2.1 Invoking FUs

The *Restart Reactivate FU* may be invoked by the following FUs:

- *Permanent Association FU*;
- *Dynamic Association FU*;
- *Restart Reconfigure FU*.

8.2.2.2.2 Invoked FUs

The *Restart Reactivate FU* may invoke the following FU:

- *Unsolicited Data Transfer FU*.

8.2.2.2.3 Disrupting FUs

The *Restart Reactivate FU* shall not be disrupted. Disruption of supporting FU invocations are treated as errors within a *Restart Reactivate FU* invocation, but do not disrupt it.

8.2.2.2.4 Disrupted FUs

The *Restart Reactivate FU* shall not disrupt any other FU.

8.2.2.3 Invocation

8.2.2.3.1 Prerequisites

No previous or concurrent FU invocations are required¹²⁸⁾.

8.2.2.3.2 Restrictions

For any given INITIATOR/RESPONDER UE combination, a *Restart Reactivate FU* invocation is not allowed, if:

- *Restart Reactivate FU* is already running for the INITIATOR/RESPONDER UE combination, or;
- *Restart Reconfigure FU* is already running for the INITIATOR/RESPONDER UE combination¹²⁹⁾.

8.2.2.3.3 Invoking events

The INITIATOR part of the *Restart Reactivate FU* may in principle be invoked whenever the INITIATOR UE determines that it is appropriate. At least, such invocation shall occur upon the following events:

- a) The restart code *Restart*, spontaneous management lost being received by the INITIATOR part of a *Permanent Association FU* or *Dynamic Association FU* invocation.

¹²⁸⁾ The *Permanent Association FU* invocation is a requirement of the *Unsolicited Data Transfer FU* invoked by the *Restart Reactivate FU*, not of the *Restart Reactivate FU* itself.

¹²⁹⁾ Consequently, any event triggering *Restart Reactivate FU* or *Restart Reconfigure FU* invocation occurring within the *Restart Reactivate FU* or *Restart Reconfigure FU* itself (triggered by a *Dynamic Association FU* or *Permanent Association FU* invocation) will have to be either ignored or deferred until the primary *Restart Reactivate FU* or *Restart Reconfigure FU* invocation is terminated.

- b) Chaque fois que la *Restart Reactivate FU* est à réinvoker en raison de la présence d'erreurs dans les FU invoquées par la *Restart Reactivate FU*, sur décision locale de l'INITIATOR UE.

La *Restart Reactivate FU* est de type composite, ce qui signifie qu'il n'existe pas, dans la *Restart Reactivate FU*, de partie RESPONDER spécifique ayant un événement invocateur spécifique associé.

8.2.2.4 Fin

8.2.2.4.1 Orderly termination

La fin ordonnée de la *Restart Reactivate FU* n'est pas définie¹³⁰⁾. Toute fin ordonnée inattendue d'une FU invoquée par la *Restart Reactivate FU* est considérée comme une erreur dans l'invocation de la *Restart Reactivate FU*.

8.2.2.4.2 Disruption

Une invocation de la *Restart Reactivate FU* ne doit pas être interrompue. Voir 8.2.2.2.3.

8.2.3 Procédures

8.2.3.1 Primitives de service de TASE.1

La *Restart Reactivate FU* est de type composite, ce qui signifie qu'aucune séquence de primitives de service de TASE.1 ne lui est associée.

8.2.3.1.1 Séquence d'actions

La séquence d'actions normale se répartit en deux phases:

Phase 1: Autorisation d'émettre accordée au RESPONDER UE

Pour chaque groupe défini dans le CS(I)¹³¹⁾:

L'INITIATOR UE examine les informations d'état locales concernant le groupe et décide s'il convient d'autoriser le RESPONDER UE à envoyer des données appartenant au groupe, spontanément ou périodiquement.

S'il convient d'accorder l'autorisation d'émettre (périodiquement ou spontanément) des données non sollicitées:

Si les conditions préalables et limites applicables à l'invocation d'une *Unsolicited Data Transfer FU* de ce groupe peuvent être respectés:

L'INITIATOR UE invoque la *Unsolicited Data Transfer FU* de ce groupe.

Si la *Unsolicited Data Transfer FU* n'est PAS en fonctionnement ou si la phase 1¹³²⁾ de l'invocation de la *Unsolicited Data Transfer FU* ne s'effectue PAS correctement:

L'invocation de la *Restart Reactivate FU* traite l'erreur.

¹³⁰⁾ Ceci signifie que toute invocation de *Restart Reactivate FU* aura à suivre son cours et se terminer d'elle-même.

¹³¹⁾ Contrairement à la phase 1 de la *Restart Reconfigure FU*, les groupes éventuellement prédéfinis dans le CS sont pris en compte ici.

¹³²⁾ Dans ce contexte, il suffit que la phase 1 de la *Unsolicited Data Transfer FU* réussisse pour que l'invocation soit réussie.

- b) Whenever re-invocation of the *Restart Reactivate FU* is to be made, because of errors in FUs invoked by the *Restart Reactivate FU*, subject to local decision in the INITIATOR UE.

The *Restart Reactivate FU* is of the composite type. Consequently, there exists no specific RESPONDER part of the *Restart Reactivate FU*, with an associated specific invoking event.

8.2.2.4 Termination

8.2.2.4.1 Orderly termination

Orderly termination of the *Restart Reactivate FU* is not defined¹³⁰⁾. Unexpected orderly termination of any FU that the *Restart Reactivate FU* invokes is considered an error within the *Restart Reactivate FU* invocation.

8.2.2.4.2 Disruption

A *Restart Reactivate FU* invocation shall not be disrupted. See 8.2.2.2.3.

8.2.3 Procedures

8.2.3.1 TASE.1 service primitives

The *Restart Reactivate FU* is of the composite type, having no specific TASE.1 service primitive sequence associated with it.

8.2.3.1.1 Action sequence

The normal action sequence has two phases:

Phase 1: Granting the RESPONDER UE permissions to send

For each group that is defined in the CS¹³¹⁾:

The INITIATOR UE examines its local state information regarding the group, and decides whether the RESPONDER UE should be granted permission to send data belonging to the group, either spontaneously or periodically.

If permission to send unsolicited data (periodically or spontaneously) should be granted:

If the prerequisites and restrictions that apply to a *Unsolicited Data Transfer FU* invocation for the group can be met:

The INITIATOR UE invokes the *Unsolicited Data Transfer FU* for the group.

If the *Unsolicited Data Transfer FU* is NOT running OK or phase 1¹³²⁾ of the *Unsolicited Data Transfer FU* invocation does NOT complete OK:

The *Restart Reactivate FU* invocation processes the error.

¹³⁰⁾ This means that any *Restart Reactivate FU* invocation will have to run its course until it terminates itself.

¹³¹⁾ Groups that are fixedly predefined in the CS, if any, are included here, in contrast to what is the case for phase 1 of the *Restart Reconfigure FU*.

¹³²⁾ OK completion of phase 1 of the *Unsolicited Data Transfer FU* is sufficient for success in this context.

Sinon:

L'invocation de la *Restart Reactivate FU* traite l'erreur¹³³⁾.

Phase 2: Traitement des erreurs

Si au moins une erreur sérieuse¹³⁴⁾ a été générée au cours de l'invocation de la *Restart Reactivate FU*, l'INITIATOR UE doit effectuer l'une des actions suivantes:

- a) Terminer l'invocation de la *Restart Reactivate FU* en signalant l'erreur pour réactiver les groupes du RESPONDER UE.¹³⁵⁾
- b) Terminer l'invocation de la *Restart Reactivate FU*. Attendre pendant un moment déterminé localement, puis recommencer l'invocation de la *Restart Reactivate FU* à partir du début.

Effectuer le cycle d'attente/réinvocation jusqu'à ce que l'invocation réussisse ou que le nombre d'invocations maximal déterminé localement soit atteint.

Si l'invocation n'est toujours pas réussie:

- terminer et signaler l'erreur, tel que décrit au point a), ci-dessus.

8.2.3.1.2 Parameter values

La FU est de type composite. Voir la description des FU en 4.2.2.

8.2.3.2 Traitement des erreurs

Les erreurs générées avec les primitives de service de TASE.1 ne sont pas étudiées ici, car la *Restart Reactivate FU* est de type composite.

Le traitement des erreurs générées dans les FU invoquées par la *Restart Reactivate FU* est expliqué en 8.2.3.1.1.

8.2.3.2.1 FU disruption

La *Restart Reactivate FU* ne peut pas être interrompue.

8.2.3.2.2 Illegal invocation attempt

FU non présente:

Aucune.

Si la *Restart Reactivate FU* n'est pas présente dans un INITIATOR UE:

Les requêtes d'invocation sont toujours générées localement; voir 8.2.2.3. En conséquence, ce type d'erreur est traité localement et ne fait pas partie du domaine d'application de ce rapport technique.

¹³³⁾ Si la *Unsolicited Data Transfer FU* est déjà en cours d'invocation pour ce groupe, ceci n'est pas considéré comme une erreur.

¹³⁴⁾ Sur décision locale de l'INITIATOR UE.

¹³⁵⁾ La suite du traitement de l'échec de réinitialisation se fait au niveau local, dans l'INITIATOR UE, et ne fait pas partie du domaine d'application du présent rapport technique. Cependant, ce type d'échec signifie que les données des groupes ayant échoué n'e seront pas transmises par le RESPONDER UE comme attendu. Il convient donc de déclencher des actions de correction, généralement manuelles.

If not:

The *Restart Reactivate FU* invocation processes the error¹³³⁾.

Phase 2: Error handling

If at least one serious¹³⁴⁾ error has been encountered during the current *Restart Reactivate FU* invocation, the INITIATOR UE shall follow one of two courses of action:

- a) Terminate the *Restart Reactivate FU* invocation, flagging failure to re-activate groups in the RESPONDER UE¹³⁵⁾.
- b) Terminate the *Restart Reactivate FU* invocation. Wait for a locally determined amount of time, then re-invoke the *Restart Reactivate FU*, from the beginning.

Perform the wait/re-invoke cycle until success or a locally determined maximum number of invocations have been performed.

If still not successful:

- Terminate, and report the error, as described in a), above.

8.2.3.1.2 Parameter values

The FU is of the composite type. See description of component FUs in 4.2.2.

8.2.3.2 Error handling

Errors occurring with TASE.1 service primitives are not considered here, since the *Restart Reactivate FU* is of the composite type.

Handling of errors occurring in FUs invoked by the *Restart Reactivate FU* is described in 8.2.3.1.1.

8.2.3.2.1 FU disruption

The *Restart Reactivate FU* cannot be disrupted.

8.2.3.2.2 Illegal invocation attempt

FU not present:

None.

If the *Restart Reactivate FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see 8.2.2.3. Consequently, the handling of this type of error is a local issue, outside the scope of this technical report.

¹³³⁾ If the *Unsolicited Data Transfer FU* is already being invoked for the group in question, this is NOT considered an error.

¹³⁴⁾ Subject to local decision of the INITIATOR UE.

¹³⁵⁾ Further processing of re-activation failure is a local issue within the INITIATOR UE, outside the scope of this technical report. However, the failure means that data for failing groups will not be transmitted by the RESPONDER UE as expected, so some sort of corrective action should eventually be triggered. Such corrective action will typically be manual.

Etant donné que la *Restart Reactivate FU* est de type composite, aucune partie RESPONDER spécifique n'est définie pour cette FU.

FU présente, mais tentative illégale:

Les tentatives d'invocation multiples simultanées de la *Restart Reactivate FU* d'une combinaison INITIATOR/RESPONDER UE donnée sont traitées localement et ne font pas partie du domaine d'application du présent rapport technique. L'INITIATOR UE doit cependant s'assurer que les échecs d'invocation de la *Restart Reactivate FU* sont correctement enregistrés. Ces échecs signifient en effet que les données attendues ne seront pas transmises par le RESPONDER UE.

Il en va de même pour les tentatives d'invocation de la *Restart Reactivate FU* pour une combinaison INITIATOR/RESPONDER UE pour laquelle la *Restart Reconfigure FU* est déjà en cours de fonctionnement.

9 Sécurité

9.1 Introduction

Afin de pouvoir transférer des informations en toute sécurité via TASE.1, il faut que toute organisation définisse une *politique de sécurité* en fonction de ses besoins. Les *services de sécurité* utilisés doivent refléter la *politique de sécurité*. Ces services sont mis en place à partir d'un ensemble de *mécanismes de sécurité*.

Le Service Element (TASE.1) ne contient aucun mécanisme de sécurité. Ceux-ci doivent en effet être résidents dans l'User Element. TASE.1 permet cependant à deux User Elements de *s'authentifier* mutuellement et d'utiliser des mécanismes de sécurité lors de l'échange d'informations.

TASE.1 fournit des facilités permettant d'utiliser un ensemble de mécanismes de sécurité. Ces facilités risquent d'imposer certaines contraintes à la politique de sécurité d'une organisation utilisant TASE.1. Cependant, à ce stade, les mécanismes de sécurité proposés sont largement suffisants pour la plupart des organisations.

Les concepts, les services de sécurité et les mécanismes de sécurité sont établis à partir de l'ISO 7498-2 qui définit l'architecture de sécurité ISO et de l'UIT-T X.509 qui définit le cadre d'authentification des services de répertoire.

9.2 Services de sécurité

Les trois services de sécurité suivants sont considérés comme importants pour les TASE.1 User Elements :

- authentification de l'entité homologue;
- intégrité de la connexion de la zone de sélection;
- confidentialité de la zone de sélection.

Utilité des services:

- Authentification de l'entité homologue: Ce service fournit la preuve que l'utilisateur d'une instance précise de communication est bien l'utilisateur déclaré.
- Intégrité de la connexion de la zone de sélection: Ce service fournit la preuve de l'intégrité d'une zone d'un Protocol Data Unit échangé avec ses entités homologues. Ce service permet de s'assurer que les données n'ont pas été modifiées durant le transfert.
- Confidentialité de la zone de sélection: Ce service est utilisé pour protéger la ou les zones d'un Protocol Data Unit contre les divulgations non autorisées.

No specific RESPONDER part is defined for the *Restart Reactivate FU* since it is of the composite type.

FU present, but attempt illegal:

Attempts of multiple simultaneous invocations of the *Restart Reactivate FU* for a given INITIATOR/RESPONDER UE combination are a local issue, outside the scope of this technical report. However, the INITIATOR UE shall ensure that *Restart Reactivate FU* failures are properly registered, since such failure signifies that expected data will not be transmitted by the RESPONDER UE.

The same applies for attempts of invoking the *Restart Reactivate FU* for an INITIATOR/RESPONDER UE combination for which the *Restart Reconfigure FU* is already running.

9 Security

9.1 Introduction

An organisation must work out a *security policy* based on its needs for a secure transfer of information using TASE.1. The *security services* needed shall reflect the security policy. These services are realised by a set of *security mechanisms*.

TASE.1 does not offer any security mechanisms as part of the Service Element (TASE.1). The security mechanisms shall reside in the User Element. However, TASE.1 enables two User Elements to *authenticate* each other and to employ security mechanisms during the information exchange.

TASE.1 offers facilities to use a set of security mechanisms. This may put some constraints on the security policy for an organisation using TASE.1. However, at the current stage the security mechanisms offered are envisaged to be more than adequate for most organisations.

The concepts, security services, and mechanisms are based on ISO 7498-2, which defines the ISO security architecture and ITU-T X.509, which defines the authentication framework for the directory services.

9.2 Security services

Three security services are regarded as important for TASE.1 User Elements. They are:

- peer entity authentication;
- selective field connection integrity;
- selective field confidentiality.

The services will provide:

- Peer entity authentication: This service provides evidence that a user in a certain instance of communication is the one claimed.
- Selective field connection integrity: This service provides proof of the integrity of one field of a Protocol Data Unit exchanged with its peer entity. The service assures that data is not changed during the transfer.
- Selective field confidentiality: This service is used to protect field(s) of a Protocol Data Unit from unauthorised disclosure.

Le service *Authentification de l'entité homologue* est considéré comme le plus important des services de sécurité. Si ce service est sélectionné, l'authentification interviendra au moment de l'établissement de l'association.

Le service *Intégrité de la connexion de la zone de sélection* peut se révéler important pour certaines organisations lors du transfert de certains types de données (par exemple les données de type Supervisory Control Data). TASE.1 fournit des fonctions permettant de négocier l'utilisation de ce service pour transférer des données utilisateurs ainsi que pour échanger des définitions de groupes et des affichages de groupes.

Le service *Confidentialité de la zone de sélection* peut être utilisé pour protéger des informations sensibles (par exemple des données économiques confidentielles) contre une lecture non autorisée. TASE.1 fournit des fonctions permettant de négocier l'utilisation de ce service lors du transfert de données utilisateur ainsi que pour échanger des définitions de groupes et des affichages de groupes.

9.3 Mécanismes de sécurité

Les mécanismes de sécurité utilisés pour fournir les services de sécurité définis dans la section précédente sont les suivants:

- Echange d'authentification: Mécanisme utilisé pour authentifier l'entité homologue.
- Authentification de zone: Mécanisme utilisé pour assurer l'intégrité de la connexion de la zone de sélection. Un code d'authentification, ajouté après les données de la zone, permet d'utiliser ce service. Le code est généré à partir de la zone sélectionnée et non à partir de l'ensemble du message.
- Cryptage: Mécanisme utilisé pour rendre confidentielle la zone de sélection.

L'échange d'authentification intervient au cours de l'établissement de l'association, par le biais d'une zone d'*information d'authentification*.

Deux modes sont disponibles: l'échange unidirectionnel ou bidirectionnel. L'échange unidirectionnel permet d'authentifier l'initiateur. Dans un échange bidirectionnel, l'initiateur est authentifié pour le répondeur et le répondeur est authentifié pour l'initiateur.

Lorsqu'un échange unidirectionnel est sélectionné, une zone d'information d'authentification est envoyée par l'initiateur au répondeur. Aucune zone d'information d'authentification n'est renvoyée par le répondeur à l'initiateur.

Lorsqu'un échange bidirectionnel est utilisé, l'initiateur envoie une zone d'information d'authentification au répondeur et le répondeur doit renvoyer une zone d'informations d'authentification à l'initiateur.

Les UE doivent négocier l'utilisation de l'authentification de zone et du cryptage. Cette négociation s'effectue au cours de l'établissement de l'association. Une *zone d'options de sécurité* est envoyée par l'initiateur au répondeur au cours de l'établissement de l'association.

La zone d'options de sécurité et la zone d'information d'authentification sont envoyées de façon transparente par le biais de TASE.1. L'authentification de zone et le cryptage sont également utilisés de façon transparente pour TASE.1.

Le code d'authentification est ajouté à la zone à authentifier. Pour TASE.1, le code d'authentification fait partie intégrante de la zone. La présence de ce code réduit l'espace disponible pour les données. Dans la plupart des cas, la longueur du code d'authentification est de 32 bits ou 64 bits. La décision de générer un code d'authentification est prise par le UE qui envoie les données.

Peer entity authentication is considered the most important security service. If peer entity authentication is selected, this will take place when an association is established.

Selective field connection integrity is likely to be important for some organisations when certain types of data are transferred, e.g. the transfer of Supervisory Control Data. TASE.1 provides functions to negotiate the use of this for the transfer of user data as well as for the exchange of group definition and group readout.

Selective field confidentiality can be used to protect sensitive information, e.g. information of economical value, from unauthorised inspection. TASE.1 provides functions to negotiate the use of this during the transfer of user data as well as for the exchange of group definition and group readout.

9.3 Security mechanisms

The security mechanisms used to offer the security services defined in the previous section are:

- Authentication exchange: Used to provide peer entity authentication.
- Field authentication: Used to provide selective field connection integrity. An authentication code appended after the data of the field is used to provide this service. The code is generated on the basis of the selected field and not the complete message.
- Encipherment: Used to provide the selective field confidentiality.

The authentication exchange will take place during the association establishment. An *authentication information field* is used.

Two modes are possible: one-way exchange and two-way exchange. One-way exchange will authenticate the initiator. When two-way exchange is used, the initiator is authenticated for the responder, and the responder is authenticated for the initiator.

When one-way exchange is selected, an authentication information field is sent by the initiator to the responder. No authentication information field is returned from the responder to the initiator.

When two-way exchange is selected, the initiator will send an authentication information field to the responder, and the responder shall return an authentication information field to the initiator.

The UEs shall negotiate the use of field authentication and encipherment. This is done during association establishment. A *security options field* is sent from the initiator to the responder during association establishment.

The security options field and the authentication information field are sent transparently through the TASE.1. The use of field authentication and encipherment is transparent for the TASE.1, as well.

The authentication code is appended to the field subject to authentication. For TASE.1, the authentication code is an integral part of the field. This will reduce the available space for data. The authentication code will in most cases be 32 bits or 64 bits in length. The generation of the authentication code is the responsibility of the UE sending the data.

Certaines zones sont cryptées. Le cryptage est effectué par le UE qui envoie les données. Le UE récepteur doit être en mesure de détecter une erreur de cryptage. Cette détection se fait habituellement grâce à une somme de contrôle qui peut être incluse dans les données cryptées ou être ajoutée après la partie de données cryptées. La zone de données ainsi obtenue, plus grande que les données d'origine, ne doit pas dépasser la longueur de zone maximale. Le décryptage doit être effectué par le UE récepteur.

9.4 Classes de sécurité et options

TASE.1 définit les quatre classes de sécurité suivantes:

- Classe 0: Aucun mécanisme de sécurité n'est utilisé (classe par défaut).
- Classe 1: Utilisation de l'échange d'authentification (unidirectionnel ou bidirectionnel).
- Classe 2: Classe 1 + utilisation d'une authentification de zone.
- Classe 3: Classe 1 + utilisation du cryptage.

La classe sélectionnée est valide pour l'association en cours d'établissement. Il est impossible d'utiliser l'authentification de zone et le cryptage pour une même association. Il est permis à un UE de traiter plusieurs associations, chaque association ayant son propre jeu de classes de sécurité.

Dans le cadre de l'établissement de l'association, l'INITIATOR UE:

- sélectionne la classe de sécurité (la sélection d'une classe de sécurité n'est pas soumise à négociation);
- sélectionne l'échange d'authentification unidirectionnel ou bidirectionnel;
- détermine si l'authentification de zone ou le cryptage doivent être utilisés;
- détermine le type de données pour lesquelles l'authentification de zone ou le cryptage seront utilisés.

Zones (paramètres) dans lesquelles l'authentification de zone ou le cryptage peuvent être utilisés:

- paramètre Objects utilisé dans les primitives *DefineGroup* et *GetGroup*;
- paramètre Data utilisé dans la primitive *CommandTransfer*;
- paramètre Data utilisé dans les primitives *RequestedDataTransfer* et *Send(Non)ConfData*;
- paramètre *MixedData* utilisé dans la primitive *SendMixedData*.

Le contenu du paramètre d'une invocation de primitive unique est soumis à une authentification de zone ou à un cryptage. La longueur de ce paramètre après génération du code d'authentification ou après le cryptage ne doit pas dépasser la taille maximale de ce paramètre.

9.5 Conseils d'utilisation des mécanismes de sécurité

Le présent rapport technique décrit l'échange d'information de sécurité au cours de l'établissement de l'association. Il ne décrit pas le contenu de la zone d'information d'authentification ni la sélection des algorithmes d'authentification de zone ou de cryptage. Ces aspects sont soumis à des conventions locales. Le présent paragraphe explique néanmoins comment il est possible d'utiliser des mécanismes de sécurité.

Le contenu de la zone d'information d'authentification dépend des autres options sélectionnées. Un *Public-key-crypto-system* (PKCS) tel que RSA ou un système cryptographique symétrique tel que DES peuvent être utilisés pour la zone d'information d'authentification¹³⁶⁾.

¹³⁶⁾ Si RSA est utilisé, la longueur de clé conseillée est de 512 bits, c'est-à-dire 64 octets.

Certain fields may be subject to encipherment. The encipherment is the responsibility of the sending UE. The receiving UE shall be able to detect an encipherment error. This is usually done with a checksum, which may be included in the data subject to encipherment or appended after the enciphered data part. The resulting data field, being larger than the original data, shall not exceed the maximum field length. Decipherment shall be done by the receiving UE.

9.4 Security classes and options

TASE.1 defines four security classes. They are:

- Class 0: No security mechanisms are used. This is the default value.
- Class 1: Authentication exchange is employed. One-way or two-way exchange may be used.
- Class 2: Class 1 + use of field authentication.
- Class 3: Class 1 + use of encipherment.

The selected class is in effect for the association being established. It is not possible to use field authentication as well as encipherment for the same association. One UE may handle several associations and each association has its individual security class set.

The INITIATOR UE will, as part of the association establishment:

- select the security class (the selection of a security class is not subject to negotiations);
- select one-way or two-way authentication exchange;
- determine if field authentication or encipherment shall be used;
- determine the data types for which field authentication or encipherment shall be used.

The fields (parameters) where field authentication or encipherment can be used are:

- the Objects parameter used in the *DefineGroup* and *GetGroup* primitives;
- the Data parameter in the *CommandTransfer* primitive;
- the Data parameter in the *RequestedDataTransfer* and *Send(Non)ConfData* primitives;
- the *MixedData* parameter in the *SendMixedData* primitive.

The content of the parameter in a single primitive invocation is subject to field authentication or encipherment. The resulting length of the parameter *after* authentication code generation or encipherment shall *not* exceed the maximum size of the parameter.

9.5 Suggested use of the security mechanisms

This technical report defines *exchange of security information during association establishment*. It is outside the scope of this technical report to determine the content of the authentication information field and select the field authentication or encipherment algorithms. These aspects are subject to local conventions. However, in this subclause it is outlined how it is *possible* to use the security mechanisms.

The content of the authentication information field depends on the other options selected. It is possible to employ a Public Key Crypto System (PKCS) such as RSA or a symmetric cryptosystem such as DES for the authentication information field.¹³⁶⁾

¹³⁶⁾ If RSA is used, the recommended key length is 512 bits, i.e. 64 octets.

Dans le cas de l'utilisation d'un système cryptographique symétrique, il est permis de crypter l'ensemble de la zone ou uniquement une partie.

Le contenu de la zone dépend de la classe de sécurité sélectionnée.

Si la classe 1 est sélectionnée (cas le plus simple), il est permis à la zone d'information d'authentification de contenir (avec utilisation d'un système cryptographique symétrique) les éléments suivants:

- id de l'INITIATOR UE
- date et heure
- numéro aléatoire
- sous-zone cryptée créée à partir de la date/heure, du numéro aléatoire et d'un mot de passe défini pour l'INITIATOR UE.

La Recommandation UIT-T X.509 indique plusieurs façons d'effectuer ces opérations.

La date et l'heure, ainsi que le numéro aléatoire sont importants. Ils permettent en effet d'éviter que la même zone d'authentification soit utilisée pour deux appels d'associations consécutifs et de réduire ainsi les risques de masquage (une entité prétend être une autre entité). Une entité ne doit surtout pas s'immiscer dans la transmission et répéter le message. Le mécanisme triple est le seul qui élimine totalement les risques de masquage.

Lorsque la classe 2 ou 3 est sélectionnée, des *clés de session* doivent être sélectionnées pour l'association. Les clés de session sont utilisées pour authentifier les zones ou crypter les données. Ces clés de session sont véhiculées dans la zone d'information d'authentification. L'échange bidirectionnel est utilisé pour offrir une clé de session dans chaque direction¹³⁷).

Lorsque l'authentification de zone est utilisée, un code d'authentification est ajouté après les données d'origine. La longueur de ce code est généralement de 32 bits ou 64 bits. Il est généré par un algorithme du type DES ou par une fonction hachage.

Le cryptage peut être utilisé de différentes manières. Il est capital de pouvoir détecter les erreurs de cryptage. Une somme de contrôle (très semblable à un code d'authentification) est ajoutée aux données. Il est permis à une telle somme de contrôle de sortir du processus de cryptage. Il est également possible d'utiliser une fonction hachage pour générer une somme de contrôle avant que le cryptage soit effectué pour l'ensemble du paramètre de données (y compris la somme de contrôle).

Lorsque l'authentification de zone et le cryptage sont utilisés, il est possible d'utiliser un chaînage cryptographique, qui permet de s'assurer de l'intégrité de la séquence de zones de données. L'option CBC (Cipher Block Chaining) de DES permet d'utiliser un tel chaînage.

9.6 Définition de la zone d'information de sécurité

La zone d'information de sécurité est utilisée pour sélectionner l'option de sécurité et véhiculer les informations d'authentification vers l'entité homologue. La zone d'information de sécurité fait partie de la zone de *données utilisateur* de la primitive *TASEBind* et est divisée en trois sous-zones:

- Longueur de la zone d'information de sécurité 2 octets
- Options de sécurité:
 - Classe de sécurité 1 octet
 - Type d'échange d'authentification 1 octet
- Valeur d'authentification

¹³⁷⁾ Si RSA est utilisé, il convient, pour l'émetteur, de crypter la clé de session en utilisant la *clé publique du récepteur* (voir l'ISO 7498-2). Il convient cependant, pour l'émetteur, de crypter l'ensemble de la zone en utilisant la *clé secrète de l'émetteur*.

When a symmetric cryptosystem is used, the whole field may be enciphered or just a part of the field.

The content of the field will depend on the security class selected.

In the simplest case for class 1 the authentication information field may contain (when a symmetric cryptosystem is used):

- INITIATOR UE id
- time stamp
- a random number
- an enciphered subfield based on the time stamp, the random number and a password of the INITIATOR UE.

Recommendation ITU-T X.509 suggests several ways to do this.

The time stamp and the random number are important to avoid the same authentication field in two subsequent association calls; this to reduce the risk of masquerade (the pretence by an entity to be a different entity). It is non-trivial for one entity to tap the transmission and replay the message. A three-way handshake is the only mechanism to completely eliminate the risk of masquerade.

When class 2 or 3 is selected, *session keys* shall be selected for the association. The session keys are used for field authentication or encipherment. These session keys are conveyed in the authentication information field. Two-way exchange is employed to provide for one session key in each direction¹³⁷).

When field authentication is used, an authentication code is appended to the original data. This code will typically be 32 bits or 64 bits. It is generated with a DES-like algorithm or a hashing function.

Encipherment can be used in several ways. One important requirement is to detect an encipherment error. A checksum (in many ways like an authentication code) is appended to the data. Such a checksum may come out of the encipherment process. An alternative is to use a hashing function to generate the checksum before encipherment is done for the whole data parameter (including the checksum).

For field authentication and encipherment, it is possible to ensure the integrity of the sequence of data fields by using cryptographic chaining. The CBC (Cipher Block Chaining) option of DES offers this.

9.6 Definition of the security information field

The *security information field* is used to select the security option and to convey the authentication information to the peer entity. The security information field is a part of the *User data field* in the *TASEBind* primitive and is divided into three subfields:

- Length of security information field 2 octets
- Security options:
 - Security class 1 octet
 - Authentication exchange type 1 octet
- Authentication value

¹³⁷ If RSA is employed, the sender should encipher the session key with the *public key of the receiver* (see ISO 7498-2). However, the sender should encipher the whole field with the *secret key of the sender*.

La longueur de la zone indique la longueur des options de sécurité et de la valeur d'authentification.

La classe de sécurité contient l'une des valeurs suivantes:

- Valeur 0 (décimal 0): Classe de sécurité 0
- Valeur 1 (décimal 1): Classe de sécurité 1
- Valeur 2 (décimal 2): Classe de sécurité 2
- Valeur 3 (décimal 3): Classe de sécurité 3

Aucune limite n'est imposée quant à la longueur de la valeur d'authentification. La longueur de la valeur d'authentification varie en fonction du type de système cryptographique utilisé et des paramètres (voir 9.5). Il convient que la longueur maximale soit déterminée localement.

La zone d'information de sécurité est parfois absente. Dans ce cas, il convient de sélectionner la classe de sécurité 0. Si la zone de valeur d'authentification est absente, la zone des options de sécurité indique la classe de sécurité 0 lorsqu'elle est envoyée par l'INITIATOR UE. Dans le cas contraire, le RESPONDER UE rejette l'appel avec Result = *authentication-failure*.

IECNORM.COM : Click to view the full PDF of IEC TS 60870-6-504:1998

The length field indicates the length of the security options and the authentication value.

The security class has one of the following values:

- Value 0 (decimal 0): Security class 0
- Value 1 (decimal 1): Security class 1
- Value 2 (decimal 2): Security class 2
- Value 3 (decimal 3): Security class 3

No limit is imposed on the length of the authentication value. The length of the authentication value will vary depending on the type of cryptosystem and the parameters included (see 9.5). The maximum length should be a local convention.

The security information field may be absent. In this case security class 0 is chosen. If the authentication value field is absent, the security options field must indicate security class 0 when sent by the INITIATOR UE. If not, the RESPONDER UE will reject the call with Result = *authentication-failure*.

IECNORM.COM : Click to view the full PDF of IEC TS 60870-6-504:1998

Annexe A (normative)

Hiérarchie d'invocation des FU

Il est permis à la *Permanent Association FU* d'invoquer: (voir 5.5.2.2.2)

- *Restart Reconfigure FU*;
- *Restart Reactivate FU*.

Il est permis à la *Dynamic Association FU* d'invoquer: (voir 5.6.2.2.2)

- *Restart Reconfigure FU*.

Il est permis à la *Test Association FU* d'invoquer: (voir 5.7.2.2.2)

(Aucune FU)

Il est permis à la *Group Configuration FU* d'invoquer: (voir 6.5.2.2.2)

- *Group Management FU*,
- *Group Definition FU*.

Il est permis à la *Group Management FU* d'invoquer: (voir 6.2.2.2.2)

(Aucune FU)

Il est permis à la *Group Definition FU* d'invoquer: (voir 6.3.2.2.2)

(Aucune FU)

Il est permis à la *Group Readout FU* d'invoquer: (voir 6.4.2.2.2)

(Aucune FU)

Il est permis à la *Requested Data Transfer FU* d'invoquer: (voir 7.1.2.2.2)

(Aucune FU)

Il est permis à la *Periodically Requested Data Transfer FU* d'invoquer: (voir 7.2.2.2.2)

- *Requested Data Transfer FU*.

Il est permis à la *Unsolicited Data Transfer FU* d'invoquer: (voir 7.3.2.2.2)

(Aucune FU)

Il est permis à la *Unsolicited Periodic Data Transfer FU* d'invoquer: (voir 7.4.2.2.2)

(Aucune FU)

Il est permis à la *Unsolicited Spontaneous Data Transfer FU* d'invoquer: (voir 7.5.2.2.2)

(Aucune FU)

Il est permis à la *Supervisory Control Data Transfer FU* d'invoquer: (voir 7.6.2.2.2)

(Aucune FU)

Annex A (normative)

FU invocation hierarchy

Permanent Association FU may invoke: (see 5.5.2.2.2)

- *Restart Reconfigure FU*;
- *Restart Reactivate FU*.

Dynamic Association FU may invoke: (see 5.6.2.2.2)

- *Restart Reconfigure FU*.

Test Association FU may invoke: (see 5.7.2.2.2)

(None)

Group Configuration FU may invoke: (see 6.5.2.2.2)

- *Group Management FU*;
- *Group Definition FU*.

Group Management FU may invoke: (see 6.2.2.2.2)

(None)

Group Definition FU may invoke: (see 6.3.2.2.2)

(None)

Group Readout FU may invoke: (see 6.4.2.2.2)

(None)

Requested Data Transfer FU may invoke: (see 7.1.2.2.2)

(None)

Periodically Requested Data Transfer FU may invoke: (see 7.2.2.2.2)

- *Requested Data Transfer FU*

Unsolicited Data Transfer FU may invoke: (see 7.3.2.2.2)

(None)

Unsolicited Periodic Data Transfer FU may invoke: (see 7.4.2.2.2)

(None)

Unsolicited Spontaneous Data Transfer FU may invoke: (see 7.5.2.2.2)

(None)

Supervisory Control Data Transfer FU may invoke: (see 7.6.2.2.2)

(None)