



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Q.755.2

(09/97)

SÉRIE Q: COMMUTATION ET SIGNALISATION

Spécifications du système de signalisation n° 7 – Gestion
du système de signalisation n° 7

**Répondeur de test du gestionnaire de
transactions**

Recommandation UIT-T Q.755.2

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE Q
COMMUTATION ET SIGNALISATION

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DES SYSTÈMES DE SIGNALISATION N° 4 ET N° 5	Q.120–Q.249
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 6	Q.250–Q.309
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R1	Q.310–Q.399
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R2	Q.400–Q.499
COMMUTATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.849
Généralités	Q.700
Sous-système transport de messages	Q.701–Q.709
Sous-système commande des connexions sémaphores	Q.711–Q.719
Sous-système utilisateur de téléphonie	Q.720–Q.729
Services complémentaires du RNIS	Q.730–Q.739
Sous-système utilisateur de données	Q.740–Q.749
Gestion du système de signalisation n° 7	Q.750–Q.759
Sous-système utilisateur du RNIS	Q.760–Q.769
Sous-système application de gestion des transactions	Q.770–Q.779
Spécification des tests	Q.780–Q.799
Interface Q3	Q.800–Q.849
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1999
RNIS À LARGE BANDE	Q.2000–Q.2999

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

RECOMMANDATION UIT-T Q.755.2

RÉPONDEUR DE TEST DU GESTIONNAIRE DE TRANSACTIONS

Résumé

La version de mars 1993 de la Recommandation Q.755 pour les testeurs de protocole concernait uniquement les testeurs de sous-système MTP. Elle se subdivise à présent en une série de Recommandations.

La Recommandation Q.755.2 définit le répondeur de test du gestionnaire de transactions (TC, *transaction capabilities*) et complète la version de mars 1993 de la Recommandation Q.755.

La présence dans le système d'un répondeur de test du gestionnaire de transactions autorise l'usage de suites de tests abstraites (ATS, *abstract test suites*) pour tester les fonctionnalités de transaction définies pour ce système, indépendamment des fonctionnalités employées par les utilisateurs des gestionnaires de transactions de transaction effectivement présents.

Source

La Recommandation UIT-T Q.755.2, élaborée par la Commission d'études 11 (1997-2000) de l'UIT-T, a été approuvée le 12 septembre 1997 selon la procédure définie dans la Résolution n° 1 de la CMNT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT.

Dans certains secteurs de la technologie de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 1998

Droits de reproduction réservés. 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'UIT.

TABLE DES MATIÈRES

	Page
1	Domaine d'application..... 1
2	Références 1
3	Symboles et abréviations..... 2
4	Généralités..... 3
5	Répondeur de test de TC 5
5.1	Architecture 5
5.2	Entité ASE utilisateur testeur du gestionnaire TC..... 6
	5.2.1 Principes généraux 6
	5.2.2 Opérations et erreurs 6
5.3	Protocole de gestion de test du gestionnaire TC 7
	5.3.1 Unités de données de protocole de gestion de test..... 7
	5.3.2 Structure des commandes..... 7
	5.3.3 Syntaxe abstraite 8
	5.3.4 Procédures 8
5.4	Opérations et erreurs du répondeur de test du gestionnaire TC 12
5.5	Unités de données de protocole de gestion d'essai 14
	Annexe A – Exemple d'utilisation répondeur de test du gestionnaire TC..... 16
	Annexe B – Réalisation de boucles au moyen du répondeur de test TC..... 19

Recommandation Q.755.2

RÉPONDEUR DE TEST DU GESTIONNAIRE DE TRANSACTIONS

(Genève, 1997)

1 Domaine d'application

La présente Recommandation définit le répondeur de test du gestionnaire de transactions, à utiliser comme dispositif d'aide pour tester le gestionnaire de transactions d'un système de signalisation n° 7 de l'UIT.

La Recommandation s'appuie sur la Recommandation Q.750 en ce qui les concerne les considérations architecturales de la relation entre le répondeur testeur et sous-système d'exploitation, de maintenance et de gestion (OMAP) du SS n° 7 [9].

La Recommandation définit un répondeur testeur simple et souple, permettant d'utiliser les suites de tests abstraites (ATS, *abstract test suites*) applicables au gestionnaire de transactions, indépendamment des utilisateurs réels des capacités de transaction, résidant dans un système à l'essai (SUT, *system under test*).

Aucune hypothèse n'est formulée quant à l'implémentation réelle de l'interface entre cette fonction et le gestionnaire de transaction.

La disponibilité d'un répondeur de test du gestionnaire de transactions de conception normalisée offre les avantages suivants:

- il permet d'écrire une suite de tests abstraite exécutable en présence de n'importe quelle implémentation du gestionnaire de transactions résidant dans un système au sein duquel le répondeur testeur est également mis en œuvre;
- il permet de tester toutes les fonctionnalités définies du répondeur testeur, indépendamment du sous-ensemble de fonctionnalités effectivement utilisées par les utilisateurs du gestionnaire de transactions et qui sont disponibles pendant qu'un équipement est testé;
- il facilite l'isolation des défauts en cours de test, du fait que la réponse appropriée à un message ou une composante TC sera indépendante de la bonne exécution d'une opération réelle d'un utilisateur du gestionnaire de transactions;
- il permet de tester la pile TC avant sa transmission à un client pour assurer la prise en charge d'une ou de plusieurs applications d'utilisateur TC;
- le répondeur testeur peut même servir à des essais d'interopérabilité pile à pile indépendamment de toute application particulière d'un utilisateur TC.

2 Références

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui de ce fait en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée.

- [1] Recommandations UIT-T X.680-X.683 (1994), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- [2] Recommandation UIT-T X.690 (1994) | ISO/CEI 8825-1:1995, *Technologies de l'information – Règles de codage de la notation de syntaxe abstraite numéro un: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- [3] Recommandations UIT-T Q.711-Q.714 (1993), *Spécifications du système de signalisation n° 7 – Description fonctionnelle du sous-système commande des connexions sémaphores (SSCS) du système de signalisation n° 7.*
- [4] Recommandations Q.771-Q.775 du CCITT (1988), *Spécifications du système de signalisation n° 7 – Description fonctionnelle du gestionnaire de transactions.*
- [5] Recommandations UIT-T Q.771-Q.775 (1993), *Description fonctionnelle du gestionnaire de transactions.*
- [6] Recommandation UIT-T X.290 (1995), *Cadre général et méthodologie des tests de conformité d'interconnexion des systèmes ouverts pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Concepts généraux.*
ISO/CEI 9646-1:1994, *Technologies de l'information – Interconnexion de systèmes ouverts – Cadre général et méthodologie des tests de conformité OSI – Partie 1: Concepts généraux.*
- [7] Recommandation UIT-T I.320 (1993), *Modèle de référence du protocole RNIS.*
- [8] Recommandation I.321 du CCITT (1991), *Modèle de référence pour le protocole du RNIS large bande et son application*
- [9] Recommandation UIT-T Q.750 (1993), *Vue d'ensemble de la gestion du système de signalisation n° 7.*

3 Symboles et abréviations

La présente Recommandation utilise les abréviations suivantes:

AE	entité d'application (<i>application entity</i>)
APDU	unité de données de protocole d'application (<i>application protocol data unit</i>)
ASE	élément de service d'application (<i>application service element</i>)
ASN.1	notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
ATS	suite de tests abstraite (<i>abstract test suite</i>)
DPC	code de point de destination (<i>destination point code</i>)
GPC	code de point générateur (<i>generator point code</i>)
IUT	instance sous test (<i>implementation under test</i>)
LME	entité de gestion de niveau (<i>level management entity</i>)
LT	testeur inférieur (<i>lower tester</i>)
MIB	base d'information de gestion (<i>management information base</i>)
MSU	unité de signal de message (<i>message signal unit</i>)
MT	testeur de sous-système MTP (<i>MTP tester</i>)
MTP	sous-système de transport de messages (SS n° 7) [(SS No. 7) message transfer part]

OPC	code de point d'origine (<i>originating point code</i>)
PC	code de point (<i>point code</i>)
PDU	unité de données de protocole (<i>protocol data unit</i>)
SAP	point d'accès au service (<i>service access point</i>)
SCCP	sous-système commande de connexion sémaphore (SS n° 7) [(SS No. 7) <i>signalling connection control part</i>]
SP	point sémaphore (<i>signalling point</i>)
SS n° 7	système de signalisation n° 7 (<i>signalling system No. 7</i>)
SUT	système à l'essai (<i>system under test</i>)
TC	gestionnaire de transactions (<i>transaction capabilities</i>)
TMP	protocole de gestion de test (<i>test management protocol</i>)
TPC	code de point d'inversion (<i>turn-around point code</i>)
TTCN	notation combinée arborescente et tabulaire (<i>tree and tabular combined notation</i>)
UT	testeur supérieur (<i>upper tester</i>)

4 Généralités

On peut se servir du testeur de protocoles, quand on fait l'essai du gestionnaire de transactions du système de signalisation n° 7, qu'il s'agisse d'essais de validation d'une forme de réalisation ou d'essais de compatibilité entre deux formes de réalisation. La principale fonction du testeur est de simuler un sous-système ou une application ordinaire, vu des TC, pour générer le trafic d'essai.

Les Recommandations I.320 [7] et I.321 [8] spécifient le modèle de référence du protocole RNIS à utiliser pour le RNIS-BE et le RNIS-LB. Elles définissent le plan usager (plan U), le plan de commande (plan C) et le plan de gestion (plan M: gestion). Les principes de la division en couches s'appliquent dans chacun de ces plans. Le plan U assure le transfert du flux de l'information d'utilisateur, avec les commandes associées. Le plan C achemine l'information relative à la commande des appels et des connexions. Le plan M est divisé en deux parties: les fonctions de gestion de couche et les fonctions de gestion de plans. La gestion de plans fournit les fonctions de gestion qui s'appliquent à un système dans son ensemble; elle assure la coordination entre tous les plans et n'est pas structurée en couches. Le plan de gestion de couche contient des entités de gestion de couche (LME, *layer management entities*) ou entités LME dont chacune fournit des fonctions de gestion se rapportant aux ressources et aux paramètres résidant dans sa propre entité de protocole. La gestion de couche traite les flux d'information relatifs à l'exploitation et à la maintenance. L'interface entre couches adjacentes dans un plan, et l'interface entre une entité LME et la couche qui lui est associée, doivent être définies en termes de primitives de service. Il est inutile de définir l'interface située entre les entités LME et la gestion des plans; cette interface dépend de la forme de réalisation.

Pour le système de signalisation n° 7, l'entité de gestion de niveau (LME, *level management entity*) est définie par analogie avec l'entité LME des Recommandations I.320 et I.321. Cela permet de tenir compte des différentes positions des limites entre les niveaux inférieurs (*lower level*) du SS n° 7 et les couches de l'OSI (par exemple la partie supérieure du sous-système MTP correspond au niveau 3 du SS n° 7, le sous-système SCCP correspond au niveau 4, mais l'un et l'autre seraient inclus dans la couche 3 si l'on applique le modèle OSI au sens strict). Pour le SS n° 7, le terme "LME" est censé signifier "entité de gestion de niveau".

Ainsi, le répondeur testeur des TC (*TC test responder*) ou testeur TT est contenu dans l'entité LME de l'entité application AE. On trouvera dans la présente Recommandation la description des procédures, des messages et des sous-structures des testeurs MT et ST. Les primitives non définies, entre la gestion des plans [base d'information de gestion (MIB, *management information base*)], et le testeur TT servent uniquement à activer/désactiver les fonctions d'essai concernées (voir la Figure 1, qui reproduit la Figure 5/Q.750).

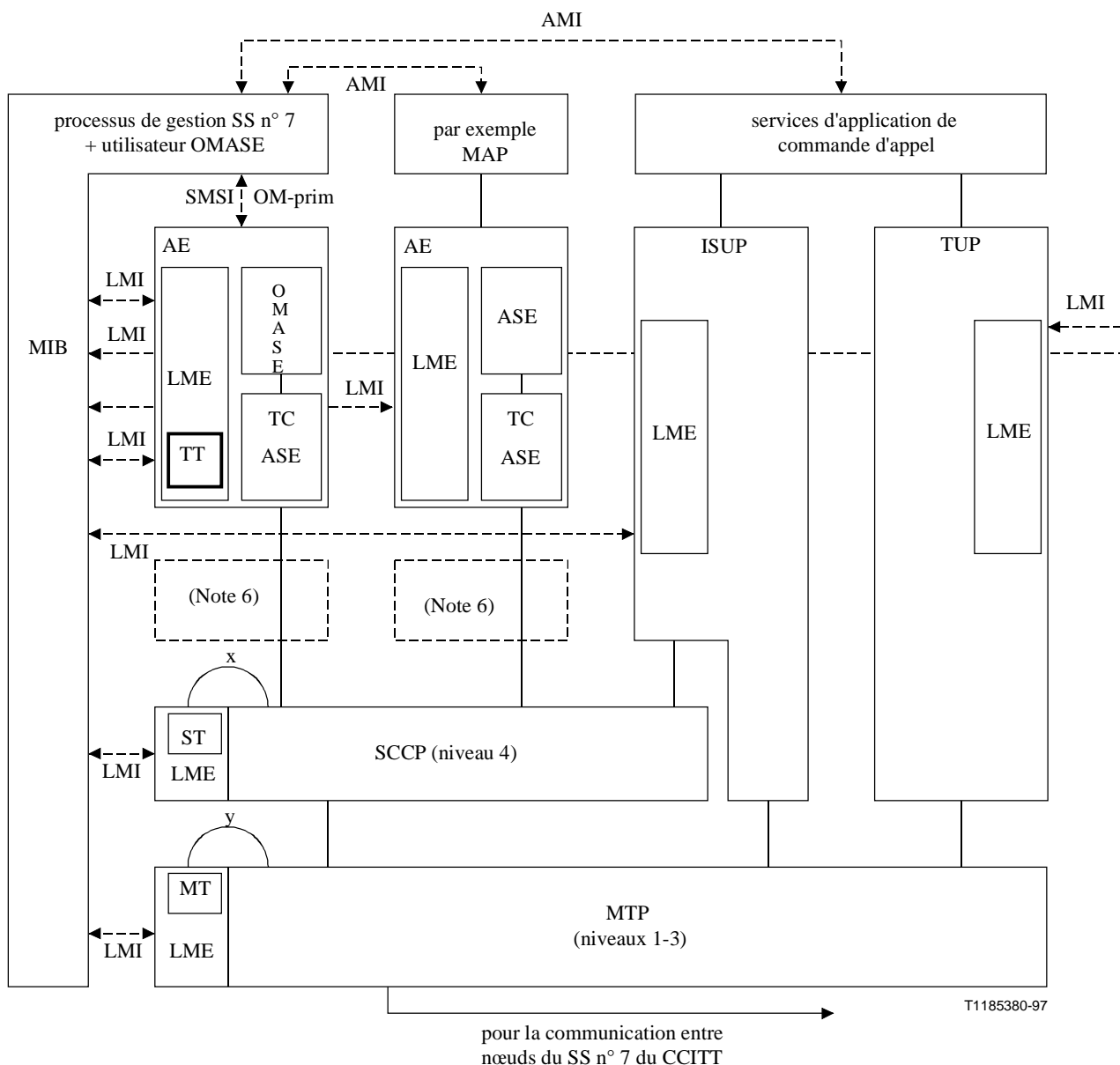


Figure 1/Q.755.2 – Gestion du SS n° 7 et configuration interne d'un point sémaphore

Des explications sont fournies à la Figure 5/Q.750 [9].

5 Répondeur de test de TC

5.1 Architecture

Le répondeur testeur est un utilisateur particulier du gestionnaire de transactions susceptible d'être mis en œuvre avec du gestionnaire de transactions au sein d'un système quelconque, suivant plusieurs configurations envisageables.

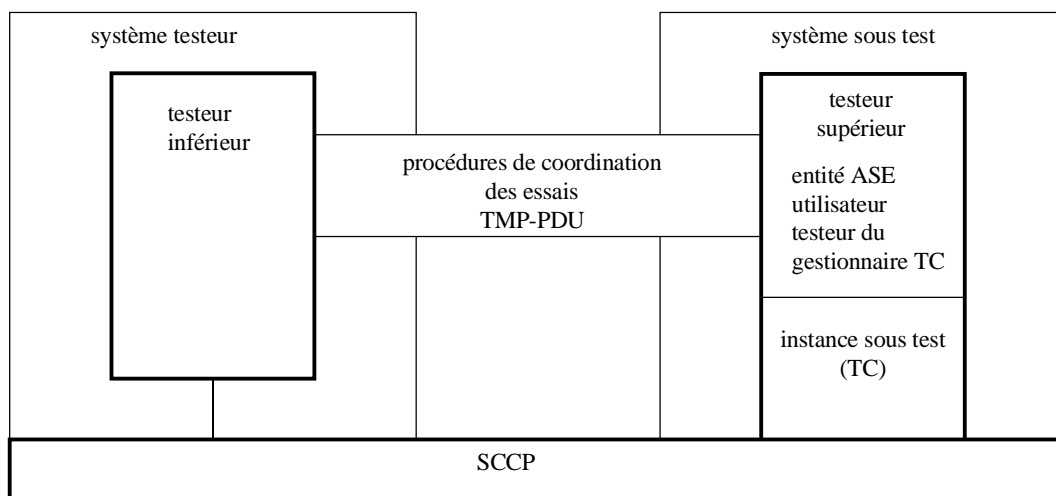
Les communications entre un répondeur testeur TC situé dans un système à l'essai (SUT, *system under test*) et un système testeur reposent sur l'utilisation d'un élément de service d'application (ASE) particulier, appelé l'ASE "utilisateur testeur du gestionnaire TC". Le répondeur testeur joue le rôle du fournisseur d'ASE, tandis que le système testeur joue celui de consommateur d'ASE.

L'ASE "utilisateur testeur du gestionnaire TC" peut être mis en œuvre en tant que composant unique d'une entité application, avec un numéro particulier de sous-système, ou peut être combiné avec d'autres éléments de la couche Application, auquel cas il suffit pour le choisir d'utiliser un nom de contexte d'application (*application-context-name*) qui commence par la valeur suivante:

{itu-t recommandation q 755 ac(5)}

Le sous-système commande de connexion sémaphore SCCP [3] attribue un numéro de sous-système pouvant servir à l'adressage d'une entité application AE contenant le répondeur de test (par exemple lorsqu'on dispose uniquement d'installations TC selon les dispositions du *Livre bleu*).

D'après la MÉTHODOLOGIE [5] DES TESTS DE CONFORMITÉ [6], L'ENSEMBLE D'UNITÉS DE DONNÉES VÉHICULÉES ENTRE LES INSTANCES DU RÉPONSEUR DE TEST ET LE SYSTÈME TESTEUR PEUT ÊTRE CONSIDÉRÉ COMME UN PROTOCOLE DE GESTION D'ESSAI INTRABANDE "IN-BAND", APPLICABLE À LA PRISE EN CHARGE DE MÉTHODES D'ESSAI COORDONNÉES ENTRE LE RÉPONSEUR DE TEST FAISANT OFFICE DE TESTEUR SUPÉRIEUR (UT, *upper tester*) ET LA FONCTIONNALITÉ DE TESTEUR INFÉRIEUR (LT, *lower tester*) DU SYSTÈME TESTEUR (VOIR LA FIGURE 2).



T1185390-97

Figure 2/Q.755.2 – Configuration de l'architecture de test

L'utilisation de ce répondeur de test n'exige aucun accès à l'interface du service TC à l'intérieur du système à l'essai, et n'impose aucune contrainte à sa mise en œuvre. Ce répondeur testeur ne fournit pas les moyens nécessaires pour tester le comportement d'une réalisation particulière du gestionnaire TC suite à un comportement non valide de l'utilisateur TC local.

Le caractère "intra bande" du protocole TMP implique que le service TC soit suffisamment fiable, bien qu'étant sous test, pour véhiculer jusqu'à un utilisateur les unités de données de protocole de ce type. Pour surmonter les difficultés potentielles dues à une perte ou à un défaut de fiabilité du service TC, les procédures ASE utilisateur testeur du gestionnaire TC sont définies de telle sorte que chaque unité TMP-PDU puisse être reçue à l'intérieur de divers types de messages ou de composantes.

Le protocole TMP a principalement pour fonction de permettre l'émission d'une série de commandes depuis le système testeur vers le répondeur testeur, de manière à déclencher un certain comportement au sein de l'instance TC sous test (IUT, *implementation under test*). Chaque commande est soit une primitive de service TC à transmettre par le répondeur testeur à l'instance TC sous test, soit une indication stipulant que le répondeur testeur doit attendre un événement ultérieur.

5.2 Entité ASE utilisateur testeur du gestionnaire TC

5.2.1 Principes généraux

L'entité ASE utilisateur testeur du gestionnaire TC peut être définie à partir de la version 1988 [4] ou de la version 1993 [5] du gestionnaire TC¹. Cette entité est capable d'entretenir simultanément plusieurs dialogues. Elle intègre les connaissances associées à sept opérations, dont deux peuvent être invoquées par le consommateur d'entité ASE (c'est-à-dire le système testeur), et cinq par le fournisseur d'entité ASE (c'est-à-dire le système sous test). Les deux opérations invoquées par le répondeur testeur sont définies en tant qu'opérations de classe 1. Toutefois cela n'a aucune incidence sur le comportement du répondeur testeur².

Les unités de données de protocole de gestion de test (TMP-PDU, *test management protocol data units*) peuvent être véhiculées dans les arguments d'opérations, les paramètres d'opérations, les paramètres d'erreur, et s'il y a lieu, dans le paramètre information utilisateur de la portion dialogue.

Il n'existe pas de relation spécifique entre une unité TMP-PDU et le type de message ou de composante utilisé pour la véhiculer.

NOTE – L'interface de service supérieure de l'entité ASE utilisateur testeur du gestionnaire TC n'est pas normalisée. Toutefois cette interface pourrait être rendue accessible localement pour déclencher un scénario de test particulier (par exemple pour des essais d'interopérabilité) ou pour permettre au répondeur testeur de signaler l'expiration d'une quelconque fonction de gestion de T-test de temporisateur.

5.2.2 Opérations et erreurs

Le module ASN.1 utilisateur testeur du gestionnaire TC défini au 5.4 contient la spécification des opérations susceptibles d'être invoquées par le système testeur ou par le répondeur testeur au cours de la communication.

Les valeurs locales attribuées aux opérations et aux erreurs suivantes sont considérées comme des valeurs par défaut. Les valeurs locales utilisées pratiquement pour ces opérations et erreurs doivent être traitées comme des paramètres de configuration.

¹ Il incombe au concepteur de la suite de test de rédiger les tests élémentaires de manière à ce que le répondeur testeur n'exige pas les fonctionnalités de la version 1993 d'une réalisation de la version de 1988 du gestionnaire TC.

² Selon les commandes reçues dans les unités de données de protocole TMP, le répondeur testeur signale ou non le résultat de ces opérations. Cela n'enfreint aucune règle en ce qui concerne le gestionnaire TC, puisque la classe d'une opération ne s'applique pas au gestionnaire TC qui résident du côté où l'opération est exécutée.

class1SupplierOperation
class2SupplierOperation
class3SupplierOperation
class4SupplierOperation
localConsumerError
localSupplierError

5.3 Protocole de gestion de test du gestionnaire TC

5.3.1 Unités de données de protocole de gestion de test

On distingue trois types d'unités de données de protocole de gestion de test (TMP-PDU)³.

Test Init

L'unité PDU TestInit demande au répondeur testeur de commencer une session de test et contient un jeu de commandes à exécuter séquentiellement par le répondeur testeur. Seul le système testeur peut envoyer ce type d'unités PDU.

Test Continue

L'unité PDU TestContinue véhicule un jeu de commandes supplémentaires à exécuter séquentiellement par le répondeur testeur. Seul le système testeur peut envoyer ce type d'unités PDU.

Test Data Echo

L'unité PDU testDataEcho véhicule des données utilisateur faisant écho aux données utilisateur reçues avec une commande particulière. Seul le répondeur testeur peut envoyer ce type d'unités PDU.

Les unités testInit et testContinue peuvent être véhiculées dans l'argument des opérations invoquées par le système testeur, dans le résultat ou le paramètre d'erreur renvoyés par le système testeur en réponse à une opération invoquée par le répondeur testeur ou encore dans le paramètre Information utilisateur de la partie dialogue des messages envoyés par le système testeur.

L'unité de données testDataEcho peut être véhiculée dans l'argument des opérations invoquées par le répondeur testeur, dans le résultat ou le paramètre d'erreur renvoyés par le système sous test en réponse à une opération invoquée par le système testeur répondeur ou encore dans le paramètre Information utilisateur de la partie dialogue des messages envoyés par le répondeur testeur.

5.3.2 Structure des commandes

Chaque commande envoyée par le système testeur dans une unité PDU testInit ou testContinue indique au répondeur testeur qu'il doit attendre un événement extérieur provenant du gestionnaire TC locales ou qu'il doit envoyer une primitive de demande particulière à destination du gestionnaire TC locales.

Les commandes notifiant au répondeur testeur d'attendre un événement extérieur précisent en outre si cet événement doit se manifester au cours d'un dialogue particulier ou si cela n'a pas d'importance.

Les commandes adressées au répondeur testeur pour lui demander d'envoyer une primitive peuvent comporter jusqu'à trois éléments:

- 1) une indication du type de primitive de demande à transmettre au gestionnaire TC locales;

³ La nécessité de définir une unité PDU de "TestEnd" doit faire l'objet d'un complément d'étude.

- 2) la référence au dialogue au cours duquel la mesure doit être prise. Si cette information n'est pas fournie explicitement, le répondeur testeur suppose que la commande se rapporte au dialogue pendant lequel l'unité de données de protocole TMP a été reçue. La référence au dialogue est toujours choisie par le système testeur. Contrairement aux identificateurs de transaction et de dialogue, la référence au dialogue est une référence commune utilisée en partage par chaque partie au dialogue;

le répondeur testeur doit garder trace de la relation biunivoque qui existe entre une référence à un dialogue et l'identificateur de dialogue du gestionnaire TC locales qui doit être convenu entre le gestionnaire TC locales et l'entité utilisateur testeur pour ce dialogue;

- 3) à titre optionnel, un paramètre de données utilisateur à renvoyer en écho en tant que données utilisateur associées à la primitive de service à émettre.

5.3.3 Syntaxe abstraite

Le module ASN.1 TC-TMP défini au 5.5 contient la spécification des unités de données de protocole de gestion d'essais.

Lorsque les unités de données de protocole TMP sont véhiculées dans le paramètre Information utilisateur de la portion dialogue, la valeur suivante de "abstract syntax name" sert de référence directe à des fins d'identification du jeu de valeurs de données, dont chacune est une valeur de type

Protocole-TMP.TMP-PDU:

`{itu-t recommendation q 755 as(4) tmp-pdus (1) version1(1)}`

Les règles de codage applicables sont les règles de codage de base définies dans la Recommandation X.690 [2].

5.3.4 Procédures

5.3.4.1 Procédures du côté du système testeur

L'entité ASE utilisateur testeur est dotée de capacités limitées de détection d'erreurs et de reprise sur erreur. Il incombe au concepteur de la suite de tests abstraite de veiller à ce que la séquence de commandes envoyées au répondeur testeur correspondent à un comportement valide d'utilisateur du gestionnaire TC et n'entrent pas en conflit avec les procédures automatiques de cette entité ASE.

Il incombe au concepteur de la suite de tests de choisir comment une unité de données de protocole TMP est véhiculée jusqu'au répondeur testeur (c'est-à-dire quel message et quelle composante le cas échéant, etc.). Le système testeur peut envoyer une unité TMP-PDU dans n'importe quel type de message ou de composante à la seule restriction qu'une unité PDU testInit ne peut être envoyée que dans un message Begin ou dans un message Unidirectional.

Il incombe en outre au concepteur de la suite de tests abstraite de veiller à ce que l'unité TMP-PDU soit effectivement communiquée à l'entité ASE utilisateur testeur (autrement dit elle doit être véhiculée dans un message ou une composante valide du gestionnaire TC).

Lancement d'un test élémentaire: au début de chaque test élémentaire, le système testeur envoie une unité de données de protocole TestInit pour assurer qu'aucune des ressources associées à un test élémentaire antérieur n'est encore active. Cette unité PDU comprend une séquence de commandes à exécuter par le répondeur testeur et, à titre optionnel, la valeur d'un délai de temporisation "chien de garde" (T-test). Faute d'indication d'un délai de temporisation, la valeur d'implémentation est choisie⁴.

⁴ Les auteurs de réalisations peuvent choisir une valeur très élevée si le répondeur testeur doit servir à des essais de trafic de base ou de charge.

Poursuite d'un test élémentaire: si toutes les commandes ne sont pas envoyées dans une unité PDU TestInit, le système testeur peut envoyer d'autres commandes au répondeur testeur au moyen de l'unité PDU testContinue, qui contient aussi une séquence de commandes à exécuter par le répondeur testeur. Un message TC peut véhiculer plusieurs unités PDU testContinue.

Fin d'un test élémentaire: il n'y a pas de commande ou d'unité TMP-PDU spécifique pour mettre fin à un test élémentaire. Il incombe au concepteur du test élémentaire de rédiger l'épilogue du test de façon à mettre un terme à tous les dialogues actifs.

5.3.4.2 Procédures du côté du répondeur testeur

5.3.4.2.1 Règles génériques

Le répondeur testeur refuse tout nom de contexte d'application (*application-context-name*) dont l'identificateur d'objet a une valeur ne commençant pas par la racine suivante:

{itu-t recommendation q 755 ac(5)}

Lorsque le répondeur testeur accepte un dialogue de type 1993 (c'est-à-dire un dialogue conforme aux indications du document [5]), il utilise le nom de contexte d'application reçu dans la prochaine primitive de gestion de dialogue envoyée.

Lorsque le répondeur testeur refuse un dialogue de type 1993, il propose la valeur suivante du nom de contexte d'application:

{itu-t recommendation q 755 ac(5) testing-ac (1) version1 (1)}

Lorsqu'il demande au gestionnaire TC d'envoyer un message Begin ou un message Unidirectional, le répondeur testeur utilise comme adresse de destination l'adresse d'origine du message dans lequel l'unité PDU testInit a été reçue. Si le répondeur testeur doit fournir une valeur de application-context-name, il donne toujours la valeur suivante:

{itu-t recommendation q 755 ac(5) testing-ac (1) version1 (1)}

Si le répondeur testeur détecte qu'il a été atteint avec une adresse de destination différente de celle qui résulte des données de configuration, il oblige le gestionnaire TC à insérer l'adresse qu'il connaît dans le premier message Continue ascendant envoyé au système testeur.

Au cours d'un dialogue, les règles suivantes s'appliquent:

- si le premier identificateur d'invocation utilisé par le répondeur testeur a la valeur 0, alors cette valeur est incrémentée à chaque invocation d'une opération par le répondeur testeur;
- si l'invocation reçue par le répondeur testeur justifie la génération d'un rejet utilisateur, le répondeur testeur demande automatiquement au gestionnaire TC de renvoyer une composante rejet (par exemple, rejet de toute opération qui n'est pas définie dans le module utilisateur testeur du gestionnaire TC) associée au problème de rejet en question;
- le répondeur testeur demande toujours l'option retour lors de l'émission d'une primitive de demande TC-BEGIN. Dans tous les autres cas, le réglage de ce paramètre est choisi lors de l'implémentation;
- lorsqu'il demande l'envoi de résultats partiels, le répondeur testeur demande toujours l'option information sur les séquences. Dans tous les autres cas, le réglage de ce paramètre est choisi lors de l'implémentation;
- la valeur du délai de temporisation fournie par le répondeur testeur lors de l'invocation d'une opération constitue un paramètre de configuration.

5.3.4.2.2 Gestion des unités de données de protocole TMP-PDU

Le répondeur testeur ne réagit pas en cas de réception d'un événement sans aucune unité TMP-PDU⁵.

Lorsqu'un événement est reçu accompagné de plusieurs unités TMP-PDU, ces dernières sont traitées séquentiellement, en commençant par celles qui sont comprises dans le champ information utilisateur de la portion dialogue (éventuelle) présente.

A réception d'une unité PDU testInit, le répondeur testeur libère toutes les ressources actives (par exemple, il met fin à tout dialogue actif, il libère les valeurs de références des dialogues, etc.)⁶ et lance le temporisateur T-test. Ensuite les commandes sont exécutées séquentiellement et le répondeur testeur attend l'événement suivant ou l'expiration du délai de temporisation T-test.

A réception d'une unité PDU testContinue, le répondeur testeur exécute séquentiellement les commandes spécifiées et attend l'événement suivant ou l'expiration du délai de temporisation T-test.

A réception d'une unité PDU testDataEcho, le répondeur testeur ne réagit pas.

Selon les règles génériques, la réception de toute autre valeur des données aura les conséquences suivantes:

- une procédure de rejet d'utilisateur-TC si la valeur non valide de l'unité PDU est reçue à l'intérieur d'un argument d'opération, d'un paramètre de résultat ou d'un paramètre d'erreur;
- une procédure d'interruption d'utilisateur-TC, si la valeur non valide est reçue à l'intérieur du champ information utilisateur de la portion dialogue.

A l'expiration de la temporisation T-test toutes les ressources associées au test sont automatiquement libérées.

Le répondeur testeur interprète et exécute séquentiellement chaque commande reçue du système testeur, tel qu'indiqué aux 5.3.4.2.3 et 5.3.4.2.4.

5.3.4.2.3 Exécution de la commande attente

Le répondeur testeur attend de l'instance sous test un événement entrant. Quand celui-ci a été reçu, il exécute la commande suivante (éventuelle) ou attend l'unité TMP-PDU suivante⁷.

En principe un événement entrant correspond à une ou plusieurs primitives d'indication TC connexes. Dans ce dernier cas le répondeur testeur reçoit une primitive de gestion de dialogue indiquant la présence de composantes. En l'occurrence le répondeur testeur consomme alors toutes les primitives de gestion de composantes, jusqu'à ce que le paramètre (dernière composante) (*last component*) prenne la valeur "TRUE". Toutefois, cela n'impose aucune contrainte à l'implémentation réelle de l'interface entre le répondeur testeur et le gestionnaire TC.

Si la commande est accompagnée d'une référence explicite à un dialogue, les primitives dont l'identificateur de dialogue ne correspondent pas à cette référence, sont ignorées (le répondeur testeur attend alors un événement suivant). Sinon, la valeur de l'identificateur de dialogue n'est pas vérifiée.

⁵ Cela comprend la réception d'une primitive d'indication TC-NOTICE susceptible de contenir une unité TMP-PDU renvoyée.

⁶ Le mode de libération des ressources est une option d'implémentation. Il ne doit cependant pas conduire à l'envoi d'un message externe quelconque (par exemple, un message END ou ABORT).

⁷ Le répondeur testeur attend toujours des événements. La commande attente est nécessaire uniquement lorsqu'il faut attendre explicitement un événement avant d'exécuter une commande envoyée auparavant.

5.3.4.2.4 Exécution d'autres commandes

Le répondeur testeur interprète le type de service comme suit:

- v1988uniReq: le répondeur testeur demande à l'instance sous test d'envoyer un message 1988 Unidirectional;
- v1993uniReq: le répondeur testeur demande à l'instance sous test d'envoyer un message 1993 Unidirectional;
- v1988beginReq: le répondeur testeur demande à l'instance sous test d'envoyer un message 1988 Begin;
- v1993beginReq: le répondeur testeur demande à l'instance sous test d'envoyer un message 1993 Begin comportant une portion dialogue;
- continueReq: le répondeur testeur demande à l'instance sous test d'envoyer un message Continue concernant le dialogue à la valeur de référencement contenue dans la commande;
- basicEndReq: le répondeur testeur demande à l'instance sous test d'envoyer un message concernant le dialogue à la valeur de référencement contenue dans la commande;
- localEndReq: le répondeur testeur demande à l'instance sous test de terminer localement le dialogue correspondant à la valeur de référencement contenue dans la commande;
- uAbortReq: le répondeur testeur demande à l'instance sous test d'interrompre le dialogue correspondant à la valeur de référencement contenue dans la commande; si le dialogue a été établi selon la procédure définie en 1993, la cause d'interruption est mise à la valeur "spécifique à l'utilisateur" (*user specific*);
- class1invokeReq: le répondeur testeur demande à l'instance sous test d'invoquer l'opération de classe 1 définie dans le module utilisateur testeur du gestionnaire TC;
- class2invokeReq: le répondeur testeur demande à l'instance sous test d'invoquer l'opération de classe 2 définie dans le module utilisateur testeur du gestionnaire TC;
- class3invokeReq: le répondeur testeur demande à l'instance sous test d'invoquer l'opération de classe 3 définie dans le module utilisateur testeur du gestionnaire TC;
- class4invokeReq: le répondeur testeur demande à l'instance sous test d'invoquer l'opération de classe 4 définie dans le module utilisateur testeur du gestionnaire TC;
- linkedInvokeReq: le répondeur testeur demande à l'instance sous test d'invoquer l'opération de classe 1 définie dans le module utilisateur testeur du gestionnaire TC en tant qu'opération liée à l'opération en instance la plus ancienne;
- resultNReq: le répondeur testeur demande à l'instance sous test d'envoyer un résultat "cette composante n'est pas la dernière" en réponse à l'opération en attente la plus ancienne;
- resultLReq: le répondeur testeur demande à l'instance sous test d'envoyer un résultat "cette composante est la dernière" en réponse à l'opération en attente la plus ancienne;
- uErrorReq: le répondeur testeur demande à l'instance sous test d'envoyer une composante erreur en réponse à l'opération en attente la plus ancienne. Le code erreur est le seul code d'erreur autorisé par la définition de l'opération;
- uCancelReq: le répondeur testeur demande à l'instance sous test d'annuler l'opération en attente la plus ancienne;
- uRejectReq: le répondeur testeur demande à l'instance sous test d'envoyer une composante rejet mentionnant "problème d'invocation: limitation de ressources" pour l'opération en attente la plus ancienne.

NOTE 1 – Dans ce contexte on entend par "message 1988", un message défini dans le document [4], et par "message 1993" un message défini dans le document [5].

NOTE 2 – Pour le module utilisateur testeur du gestionnaire TC, voir 5.4.

5.3.4.2.5 Echo de données

Lorsqu'il demande au gestionnaire de transaction 1993 un service de gestion de dialogue (voir document [5]), et en présence d'un élément "à renvoyer en écho" (*to-be-echoed*) dans la commande associée reçue du système testeur, le répondeur testeur inclut une ou plusieurs unités TMP-PDU testDataEcho en tant que valeur affectée au paramètre information utilisateur. Chaque valeur "testDataEcho" est identique à la valeur "toBeEchoed" reçue. Pendant la phase d'établissement du dialogue ou en cas d'utilisation du mode de dialogue non structuré, le nombre de valeurs de "testDataEcho" fournies par le répondeur testeur est un paramètre de configuration. Une fois le dialogue établi, seule une valeur peut être envoyée.

Lorsqu'il demande au gestionnaire TC un service de gestion de composante, le répondeur testeur inclut une unité TMP-PDU testDataEcho en tant que paramètre utilisateur (c'est-à-dire argument d'opération, paramètre de résultat, paramètre d'erreur, etc.) en présence d'un élément "à renvoyer en écho" (*to-be-echoed*) dans la commande associée reçue du système testeur. Si elle est incluse la valeur "testDataEcho" est identique à la valeur "toBeEchoed" reçue.

Si le répondeur testeur reçoit une information inconnue quelconque contenue dans le paramètre utilisateur d'une primitive de gestion de dialogue, il la fournit telle quelle dans le paramètre information utilisateur de la prochaine primitive de gestion de dialogue qu'il envoie.

5.4 Opérations et erreurs du répondeur de test du gestionnaire TC

Le module suivant définit l'entité ASE utilisateur testeur du gestionnaire TC en fonction de laquelle le système consommateur (système testeur) et le système fournisseur (système à l'essai) peuvent invoquer des opérations.

```
TC-Testing-User {itu-t recommendation q 755 modules(0) testing-user(1) version1(1)}
```

```
DEFINITIONS ::=
```

```
BEGIN
```

```
IMPORTS
```

```
OPERATION, ERROR
```

```
FROM TCAPMessages {itu-t recommendation q 773 modules(2) messages(1) version2(2)}
```

```
APPLICATION-SERVICE-ELEMENT
```

```
FROM Notation-Extensions {joint-iso-ccitt remote-operations(4) notation-extension(2)}
```

```
TMP-PDU
```

```
FROM TC-TMP {itu-t recommendation q 755 modules(0) tmp(2) version1(1)}
```

```
;
```

```
-- nom de contextes d'application
```

```
ac-id OBJECT IDENTIFIER ::= {itu-t recommendation q 755 ac(5) }
```

```
testing-ac-id OBJECT IDENTIFIER ::= {ac-id testing-ac(1) version1(1)}
```

```
-- ase
```

```
Testing-User-ASE APPLICATION-SERVICE-ELEMENT
```

```
CONSUMER INVOKES {
```

```
    localConsumerOperation,
```

```
    globalConsumerOperation
```

```
}
```

```

-- le consommateur est le système de test
SUPPLIER INVOKES {
    class1SupplierOperation,
    class2SupplierOperation,
    class3SupplierOperation,
    class4SupplierOperation,
    globalSupplierOperation
}
-- le fournisseur est le système répondeur

::= {itu-t recommendation q 755 ase(3) testing-user(1) version1(1)}

```

```

LocalConsumerOperation ::= OPERATION
ARGUMENT                TMP-PDU
RESULT                  TMP-PDU
ERRORS                  {localSupplierError}

```

```

GlobalConsumerOperation ::= OPERATION
ARGUMENT                TMP-PDU
RESULT                  TMP-PDU
ERRORS                  {globalSupplierError}

```

```

Class1SupplierOperation ::= OPERATION
ARGUMENT                TMP-PDU
RESULT                  TMP-PDU
ERRORS                  {localConsumerError}
LINKED                  {localConsumerOperation}

```

```

Class2SupplierOperation ::= OPERATION
ARGUMENT                TMP-PDU
ERRORS                  {localConsumerError}
LINKED                  {localConsumerOperation}

```

```

Class3SupplierOperation ::= OPERATION
ARGUMENT                TMP-PDU
RESULT                  TMP-PDU

```

```

Class4SupplierOperation ::= OPERATION
ARGUMENT                TMP-PDU

```

```

GlobalSupplierOperation ::= OPERATION
ARGUMENT                TMP-PDU
RESULT                  TMP-PDU
ERRORS                  {globalConsumerError}
LINKED                  {globalConsumerOperation}

```

```

ConsumerError ::= ERROR
PARAMETER        TMP-PDU

```

```

SupplierError ::= ERROR
PARAMETER        TMP-PDU

```

```

localConsumerOperation LocalConsumerOperation ::= localValue : 0

```

```

globalConsumerOperation GlobalConsumerOperation ::=
    globalValue : {itu-t recommendation q 755 operations(1) consumer(1)}

class1SupplierOperation Class1SupplierOperation ::= localValue : 1
class2SupplierOperation Class2SupplierOperation ::= localValue : 2
class3SupplierOperation Class3SupplierOperation ::= localValue : 3
class4SupplierOperation Class4SupplierOperation ::= localValue : 4
globalSupplierOperation Class1SupplierOperation ::=
    globalValue : {itu-t recommendation q 755 operations(1) supplier(2)}

localConsumerError ConsumerError ::= localValue : 1
globalConsumerError ConsumerError ::=
    globalValue : {itu-t recommendation q 755 errors(2) consumer(1)}

localSupplierError SupplierError ::= localValue : 2
globalSupplierError SupplierError ::=
    globalValue : {itu-t recommendation q 755 errors(2) supplier(2)}

```

END

5.5 Unités de données de protocole de gestion d'essai

Le module suivant définit les unités de données de protocole de gestion d'essai.

TC-TMP {itu-t recommendation q 755 modules(0) tmp(2) version1(1)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

TMP-PDU ::= CHOICE

```

    {
        testInit           [0] TestInit,
        testContinue    [1] CommandSequence,
        testDataEcho    [2] UserData
    }

```

CommandSequence ::= SEQUENCE SIZE(0..maxNbOfCommands) OF TestCommand

maxNbOfCommands INTEGER ::= 30

TestInit ::= SEQUENCE

```

    {
        timeout          INTEGER (1..127) OPTIONAL-- T-Test (unité = 30 s)
        commands        CommandSequence,
        ...
    }

```

UserData ::= CHOICE{

```

    simple      OCTET STRING (SIZE(0..maxUserDataLength)),
    complex    [0] ABSTRACT-SYNTAX.&Type
}

```

maxUserDataLength INTEGER ::= 2048

TestCommand ::= CHOICE

```
{
  wait      [0] DialogueReference,
  action    [1] ActionInfo
}
```

DialogueReference ::= CHOICE

```
{
  unspecified  NULL,
  dialogue    INTEGER (0..255)
}
```

ActionInfo ::= SEQUENCE

```
{
  service                ServiceType,
  dialogueReference      DialogueReference  DEFAULT unspecified : NULL,
  to-be-echoed          UserData  OPTIONAL,
  ...
}
```

ServiceType ::= ENUMERATED

```
{
  v1988uniReq (10),
  v1993uniReq (11),
  v1988beginReq (12),
  v1993beginReq (13),
  continueReq (14),
  basicEndReq (15),
  localEndReq (16),
  uAbortReq (17),
  class1invokeReq (21),
  class2invokeReq (22),
  class3invokeReq (23),
  class4invokeReq (24),
  linkedInvokeReq (25),
  resultNReq (26),
  resultLReq (27),
  uErrorReq (28),
  uCancelReq (29),
  uRejectReq (30),
  ...
}
```

-- Nom de syntaxe abstraite des TMP-PDU

tmp-pdus-as OBJECT IDENTIFIER ::= {itu-t recommendation q 755 as(4) tmp-pdus(1) version1(1)}

END

ANNEXE A

Exemple d'utilisation répondeur de test du gestionnaire TC

La présente annexe montre comment l'utilisation du répondeur testeur permet de répondre à certains des objectifs des tests définis dans la Recommandation Q.787.

a) Test 2.1.6: fonctions valides, annulation utilisateur

Interface répondeur testeur/TC	Flux de messages	Système testeur
	BEGIN [Invoke (localConsumerOperation, 1)] ←-----	
TC-Begin-Ind TC-Invoke-Ind (1)		
TC-Invoke-Req (0) TC-ContinueReq	CONTINUE[Invoke (class1SupplierOperation, 0)] ----->	
TC-U-Cancel-Req (0)		
	CONTINUE[Return-Result-L (0)] ←-----	
TC-L-Reject-Ind (0)		
TC-End-Req		
	END[Reject (0)]	

L'argument localConsumerOperation est défini comme suit:

```

testInit : {
    timeOut 30,
    commands
    {
        action : {service class1InvokeReq},
        action : {service continueReq},
        action : {service uCancelReq},
        wait   : {unspecified : NULL},
        action : {service basicEndReq}
    }
}
    
```

b) **Test 2.1.2.1.1: fonctions valides, opérations liées, opération originale classe 1, instance sous test émettrice**

Interface répondeur testeur/TC	Flux de messages	Système testeur
	BEGIN [Invoke(localConsumerOperation, 1)] ←-----	
TC-Begin-Ind TC-Invoke-Ind(1)		
TC-Invoke-Req (0) TC-Continue-Req		
	CONTINUE[INV(class1SupplierOperation, 0)] ----->	
	CONTINUE[Invoke(localConsumerOperation, 2)] ←-----	
TC-Continue-Ind TC-Invoke-Ind (2)		
TC-Result-L(2) TC-Continue-Req		
	CONTINUE[Return-Result-L(2)]	
	END[Return-Result-L(0)]	
TC-End-Ind TC-Result-L(0)		

Le premier argument d'invocation localConsumerOperation est défini comme suit:

```

testInit : {
    timeOut 30,
    commands {
        action : {service class1InvokeReq},
        action : {service continueReq},
        wait : {unspecified : NULL}
    }
}

```

Le deuxième argument d'invocation localConsumerOperation est défini comme suit:

```

testContinue : {
    action : {service resultLReq},
    action : {service continueReq},
    wait : {unspecified : NULL},
}

```

c) **Test 1.1.2.2.1.1-3: fonctions valides, libération suite au message Continue, interruption de l'instance sous test par utilisateur TC**

Interface répondeur testeur/TC	Flux de messages	Système testeur
	BEGIN [Invoke(localConsumerOperation, 1)] <-----	
TC-Begin-Ind (0) TC-Invoke-Ind (1)		
TC-Begin-Req (1)		
	BEGIN ----->	
	CONTINUE <-----	
TC-Continue-Ind (1)		
TC-U-Abort-Req(1)		
	ABORT ----->	
TC-End-Req(local,0)		

L'argument d'invocation localConsumerOperation est défini comme suit:

```

testInit : {
    timeOut 30,
    commands {
        action : {service v1988beginReq, dialogueReference value :1},
        wait   : {dialogueReference value :1},
        action : {service uAbortReq, dialogueReference value :1},
        action : {service localEndReq, dialogueReference value : 0}
    }
}

```


ANNEXE B

Réalisation de boucles au moyen du répondeur de test TC

La présente annexe illustre l'utilisation du répondeur testeur pour réaliser des boucles. Le test est conçu de telle sorte que le système testeur et le répondeur testeur continuent indéfiniment à amorcer des dialogues auxquels l'autre entité met fin immédiatement.

Interface répondeur testeur/TC	Flux de messages	Système testeur
	BEGIN (OTID = X1, testInit) <-----	
TC-Begin-Ind (0)		
TC-Begin-Req (1)		
	BEGIN (OTID = Y1) ----->	
TC-End-Req (0)	END (DTID = X1) ----->	
	END (DTID = Y1) <-----	
	BEGIN (OTID = X2, testContinue) <-----	
TC-Begin-Ind (2)		
TC-Begin-Req (3)		
	BEGIN (OTID = Y2) ----->	
TC-End-Req (2)	END (DTID = X2) ----->	
	END (DTID = Y2) <-----	
...		

Le premier message BEGIN envoyé du système testeur au répondeur testeur contient une unité de données de protocole testInit définie comme suit:

```

testInit : {
    timeOut 30,
    commands {
        action : {service v1988beginReq, dialogueReference value: 1},
        action : {service basicEndReq, dialogueReference value: 0},
        wait   : {dialogueReference value: 1}
    }
}
    
```

Les messages BEGIN suivants contiennent une unité de données de protocole testContinue définie comme suit:

```
testContinue : {  
    action : {service v1988beginReq, dialogueReference value: i+1},  
    action : {service basicEndReq, dialogueReference value: i},  
    wait : {dialogueReference value : i+1}  
}
```

avec i incrémenté depuis la valeur 1 jusqu'au nombre N de boucles requis.

Lorsque $i = N$, il est possible de mettre fin au test en envoyant une unité PDU testContinue sans la commande v1988beginReq. Dans cet exemple il y a au maximum deux dialogues simultanés. Toutefois lorsqu'il s'agit de tester la charge, il est possible de réécrire le test de manière à ce que le système testeur n'envoie aucun message END et que la liste des commandes des unités TMP-PDU (sauf l'unité qui marque la fin du test) ne contienne aucune commande "basicEndRequest".

SERIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Z	Langages de programmation