



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.737

(11/95)

**RÉSEAUX DE DONNÉES ET COMMUNICATION
ENTRE SYSTÈMES OUVERTS
GESTION OSI**

**TECHNOLOGIES DE L'INFORMATION –
INTERCONNEXION DES SYSTÈMES
OUVERTS – GESTION-SYSTÈMES:
CATÉGORIES DE TESTS DE CONFIANCE
ET DE DIAGNOSTIC**

Recommandation UIT-T X.737

(Antérieurement «Recommandation du CCITT»)

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. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT), (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

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. Le texte de la Recommandation X.737 de l'UIT-T a été approuvé le 21 novembre 1995. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10164-14.

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.

© UIT 1997

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.

RECOMMANDATIONS UIT-T DE LA SÉRIE X
**RÉSEAUX DE DONNÉES ET COMMUNICATION
ENTRE SYSTÈMES OUVERTS**

(Février 1994)

ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X

Domaine	Recommandations
RÉSEAUX PUBLICS POUR DONNÉES	
Services et services complémentaires	X.1-X.19
Interfaces	X.20-X.49
Transmission, signalisation et commutation	X.50-X.89
Aspects réseau	X.90-X.149
Maintenance	X.150-X.179
Dispositions administratives	X.180-X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200-X.209
Définition des services	X.210-X.219
Spécifications des protocoles en mode connexion	X.220-X.229
Spécifications des protocoles en mode sans connexion	X.230-X.239
Formulaires PICS	X.240-X.259
Identification des protocoles	X.260-X.269
Protocoles de sécurité	X.270-X.279
Objets gérés de couche	X.280-X.289
Test de conformité	X.290-X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Considérations générales	X.300-X.349
Systèmes mobiles de transmission de données	X.350-X.369
Gestion	X.370-X.399
SYSTÈMES DE MESSAGERIE	X.400-X.499
ANNUAIRE	X.500-X.599
RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES	
Réseautage	X.600-X.649
Dénomination, adressage et enregistrement	X.650-X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680-X.699
GESTION OSI	X.700-X.799
SÉCURITÉ	X.800-X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850-X.859
Traitement des transactions	X.860-X.879
Opérations distantes	X.880-X.899
TRAITEMENT OUVERT RÉPARTI	X.900-X.999

TABLE DES MATIÈRES

		<i>Page</i>
1	Domaine d'application.....	1
2	Références normatives	1
	2.1 Recommandations Norme internationales identiques	1
	2.2 Paires de Recommandations Normes internationales équivalentes par leur contenu technique	2
	2.3 Références additionnelles	2
3	Définitions.....	3
	3.1 Définitions du modèle de référence de base	3
	3.2 Définitions du cadre de gestion.....	3
	3.3 Définitions du service CMIS.....	3
	3.4 Définitions relatives à l'aperçu général de la gestion-système	3
	3.5 Définitions du modèle d'informations de gestion.....	3
	3.6 Définitions des formulaires ICS de gestion	3
	3.7 Définitions relatives à la gestion des tests.....	4
	3.8 Définitions relatives aux tests de conformité OSI.....	4
	3.9 Définitions additionnelles	4
4	Symboles et abréviations.....	5
5	Conventions.....	5
6	Prescriptions	5
7	Modèle.....	6
	7.1 Test de connexion	8
	7.2 Test de connectivité	10
	7.3 Test d'intégrité des données	12
	7.4 Test de bouclage	14
	7.5 Test d'intégrité de protocole.....	19
	7.6 Test de limite de ressource.....	22
	7.7 Autotest de ressource	25
	7.8 Test d'infrastructure d'essai	27
8	Définitions génériques	29
	8.1 Types d'attributs génériques.....	29
	8.2 Objets gérés.....	31
	8.3 Définition générique importée	32
	8.4 Compatibilité.....	32
9	Services	32
10	Unités fonctionnelles.....	32
11	Protocole	33
12	Relations avec d'autres fonctions	33
13	Conformité	33
	13.1 Conformité statique.....	33
	13.2 Conformité dynamique	33
	13.3 Prescriptions relatives à la déclaration de conformité d'instance de gestion.....	33

	<i>Page</i>
Annexe A – Définition de l'information	34
Annexe B – Formulaire MCS	54
Annexe C – Formulaire MICS.....	55
Annexe D – Formulaire MOCS.....	56
Annexe E – Formulaire MRCS	57
Annexe F – Récapitulatif des catégories de tests.....	58
Annexe G – Exemple de paquetage TARR pour le test de connexion	59
Annexe H – Catégorie de test de limite de ressource	61
Annexe I – Exemple de processus de test pour un réseau de communication géré.....	63

Résumé

La présente Recommandation | Norme internationale spécifie un certain nombre de «catégories de test» généralement utiles telles que les catégories de test de «demande de connexion», de bouclage et d'intégrité des données. Pour chacune de ces catégories, on utilise une méthode normalisée en fonction des composantes de la catégorie de tests telles que le but du test, les ressources définies dans la Recommandation X.745, l'environnement du test et le contenu du compte rendu du résultat, etc. La présente Recommandation définit la relation «d'héritage» entre les classes d'objets «tests» (qui représentent les catégories de test) et inclut les spécifications de ces classes d'objets en tant qu'objets gérés. La présente Recommandation | Norme internationale offre un moyen à d'autres groupes, notamment aux groupes associés au RGT, d'utiliser la même méthode pour la spécification des catégories de test. Cette Recommandation | Norme internationale doit être prise en considération conjointement avec la Recommandation X.745 sur la fonction de gestion des tests.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES
OUVERTS – GESTION-SYSTÈMES: CATÉGORIES DE TESTS
DE CONFIANCE ET DE DIAGNOSTIC**

1 Domaine d'application

La présente Recommandation | Norme internationale définit une fonction de gestion-système qui pourra être utilisée par un processus d'application dans un environnement de gestion centralisée ou répartie en vue d'interactions relatives à la gestion-système telle que définie par la Rec. X.700 du CCITT | ISO/CEI 7498-4. La présente Recommandation | Norme internationale définit une fonction qui se compose de définitions génériques, de services et d'unités fonctionnelles. Cette fonction se situe dans la couche Application selon la Rec. UIT-T X.200 | ISO/CEI 7498-1. Elle est définie conformément au modèle décrit par la Rec. UIT-T X.207 | ISO/CEI 9545. Le rôle des fonctions de gestion-système est décrit par la Rec. X.701 du CCITT | ISO/CEI 10040.

La présente Recommandation | Norme internationale:

- définit les besoins des utilisateurs aux fins de la présente Recommandation | Norme internationale;
- spécifie des catégories de tests de confiance et de diagnostic se rapportant aux définitions génériques indiquées par la présente Recommandation | Norme internationale selon les besoins des utilisateurs;
- définit les objets gérés, les blocs, les types d'attribut, les types d'opération et les paramètres dont les détails figurent dans la Rec. X.722 du CCITT | ISO/CEI 10165-4;
- spécifie les règles de conformité qui devront être suivies par d'autres Recommandations | Normes faisant usage des définitions génériques de la présente Recommandation | Norme internationale.

La présente Recommandation | Norme internationale:

- ne définit pas la nature d'une quelconque instance destinée à effectuer des tests de confiance et de diagnostic;
- ne spécifie pas la manière selon laquelle la gestion est effectuée par l'utilisateur des tests de confiance et de diagnostic;
- ne définit pas la nature de quelconques interactions aboutissant à utiliser des tests de confiance et de diagnostic;
- ne spécifie pas les services nécessaires pour l'établissement d'une association de gestion ni pour sa dissolution, normale ou anormale.

2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes internationales sont sujettes à révision, et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes internationales indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT-T tient à jour une liste des Recommandations UIT-T en vigueur.

2.1 Recommandations | Norme internationales identiques

- Recommandation UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: le modèle de référence de base.*

- Recommandation UIT-T X.207 (1993) | ISO/CEI 9545:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de la couche Application.*
- Recommandation UIT-T X.210 (1993) | ISO/CEI 10731:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: conventions pour la définition des services OSI.*
- Recommandation X.701 du CCITT (1992) | ISO/CEI 10040:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Aperçu général de la gestion des systèmes.*
- Recommandation X.720 du CCITT (1992) | ISO/CEI 10165-1:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: modèle d'information de gestion.*
- Recommandation X.721 du CCITT (1992) | ISO/CEI 10165-2:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: définition des informations de gestion.*
- Recommandation X.722 du CCITT (1992) | ISO/CEI 10165-4:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: directives pour la définition des objets gérés.*
- Recommandation UIT-T X.724 (1993) | ISO/CEI 10165-6:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de l'information de gestion: spécifications et directives pour l'établissement des formulaires de déclaration de conformité d'instances associés à la gestion OSI.*
- Recommandation UIT-T X.745 (1993) | ISO/CEI 10164-12:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion des systèmes: fonction de gestion de tests.*

2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique

- Recommandation X.208 du CCITT (1988), *Spécification de la syntaxe abstraite numéro un (ASN.1).*
ISO/CEI 8824:1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de la notation de syntaxe abstraite numéro 1 (ASN.1).*
- Recommandation UIT-T X.290 (1995), *Cadre général et méthodologie des tests de conformité OSI 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.*
- Recommandation UIT-T X.291 (1995), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Spécification de suite de tests abstraite.*
ISO/CEI 9646-2:1994, *Technologies de l'information – Interconnexion de systèmes ouverts – Cadre général et méthodologie des tests de conformité OSI – Partie 2: Spécification des suites de tests abstraites.*
- Recommandation X.700 du CCITT (1992), *Cadre de gestion pour l'interconnexion des systèmes ouverts pour les applications du CCITT.*
ISO/CEI 7498-4:1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 4: Cadre général de gestion.*
- Recommandation X.710 du CCITT (1991), *Définition du service commun de transfert d'informations de gestion pour les applications du CCITT.*
ISO/CEI 9595:1991, *Technologies de l'information – Interconnexion de systèmes ouverts – Définition du service commun d'informations de gestion.*

2.3 Références additionnelles

- Recommandation O.151 du CCITT (1992), *Appareil pour la mesure du taux d'erreur fonctionnant au débit primaire et au-dessus.*
- Recommandation O.152 du CCITT (1992), *Appareil de mesure du taux d'erreur pour les débits de 64 kbit/s et $N \times 64$ kbit/s.*
- Recommandation UIT-T V.32 (1993), *Famille de modems à deux fils fonctionnant en duplex à des débits binaires allant jusqu'à 9600 bit/s pour usage sur le réseau téléphonique général avec commutation et sur les circuits loués de type téléphonique.*

3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes sont utilisées.

3.1 Définitions du modèle de référence de base

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. UIT-T X.200 | ISO/CEI 7498-1:

- a) système ouvert;
- b) gestion-système.

3.2 Définitions du cadre de gestion

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.700 du CCITT | ISO/CEI 7498-4:

- a) objet géré;
- b) information de gestion.

3.3 Définitions du service CMIS

La présente Recommandation | Norme internationale utilise le terme suivant, défini dans la Rec. X.710 du CCITT | ISO/CEI 9595:

- attribut.

3.4 Définitions relatives à l'aperçu général de la gestion-système

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.701 du CCITT | ISO/CEI 10040:

- a) agent;
- b) rôle d'agent;
- c) définitions génériques;
- d) classe d'objets gérés;
- e) déclaration de conformité d'objet géré;
- f) système géré;
- g) déclaration de conformité d'information de gestion;
- h) gestionnaire;
- i) rôle de gestionnaire;
- j) formulaire MICS;
- k) formulaire MOCS;
- l) notification.

3.5 Définitions du modèle d'informations de gestion

La présente Recommandation | Norme internationale utilise le terme suivant, défini dans la Rec. X.720 du CCITT | ISO/CEI 10165-1:

- type d'attribut.

3.6 Définitions des formulaires ICS de gestion

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. UIT-T X.724 | ISO/CEI 10165-6:

- a) déclaration de conformité de relation gérée;
- b) récapitulatif de conformité de gestion;

ISO/CEI 10164-14 : 1996 (F)

- c) formulaire MCS;
- d) formulaire MRCS.

3.7 Définitions relatives à la gestion des tests

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. UIT-T X.745 | ISO/CEI 10164-12:

- a) objets associés;
- b) test piloté;
- c) objet géré renvoyant à un test (objet MORT);
- d) test intrusif;
- e) test non intrusif;
- f) test non piloté;
- g) récepteur de demande d'action de test (récepteur TARR);
- h) conducteur du test;
- i) appel de test;
- j) objet «test» (objet TO);
- k) automate de test;
- l) demande de test;
- m) compte rendu sollicité;
- n) temporisation d'exécution;
- o) compte rendu non sollicité.

3.8 Définitions relatives aux tests de conformité OSI

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans les Rec. UIT-T X.290 | ISO/CEI 9646-1 et UIT-T X.291 | ISO/CEI 9646-2:

- a) formulaire PICS;
- b) point de commande et d'observation;
- c) déclaration de conformité d'une instance de protocole;
- d) déclaration de conformité d'un système.

3.9 Définitions additionnelles

3.9.1 temps d'établissement: durée nécessaire pour établir une connexion entre deux entités de protocole.

3.9.2 terminaison explicite: terminaison d'un test à la demande du conducteur de test.

3.9.3 compte rendu implicite: envoi des résultats sous la forme d'un rapport d'événement (compte rendu non sollicité) ou d'une réponse d'action.

3.9.4 terminaison implicite: terminaison d'un test sur achèvement ou sur erreur.

3.9.5 temporisation aller-retour: délai durant lequel le conducteur de test attendra le retour des données transmises.

3.9.6 temps de propagation aller-retour: délai associé à la transmission d'une trame vers le point de bouclage et retour.

3.9.7 durée de trame: délai de transmission de chaque trame de test.

3.9.8 instant de départ: heure programmée pour le déclenchement du test.

3.9.9 espacement des tests: délai séparant la fin de la réception d'une trame d'essai et le début de l'émission de la suivante.

3.9.10 trame de test: signaux ou données à injecter sur le trajet de communication.

4 Symboles et abréviations

AO	Objet associé (<i>associated object</i>)
ASN.1	Notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
LLC	Commande de liaison logique (<i>logical link control</i>)
MCS	Résumé de conformité de gestion (<i>management conformance summary</i>)
MICS	Déclaration de conformité d'information de gestion (<i>management information conformance statement</i>)
MIDS	Déclaration de définition d'information de gestion (<i>management information definition statement</i>)
MOCS	Déclaration de conformité d'objet géré (<i>managed object conformance statement</i>)
MORT	Objet géré renvoyant à un test (<i>managed object referring to test</i>)
MRCS	Déclaration de conformité d'objet géré (<i>managed relationship conformance statement</i>)
OSI	Interconnexion des systèmes ouverts (<i>open systems interconnection</i>)
PCO	Point de contrôle et d'observation (<i>point of control and observation</i>)
PDU	Unité de données protocolaire (<i>protocol data unit</i>)
PICS	Déclaration de conformité d'instance de protocole (<i>protocol implementation conformance statement</i>)
TARR	Récepteur de demande d'action de test (<i>test action request receiver</i>)
TO	Objet «test» (<i>test object</i>)
XID	Trame d'échange d'identifications (<i>exchange identification frame</i>)

5 Conventions

La présente Recommandation | Norme internationale ne définit pas de convention particulière.

6 Prescriptions

La présente Recommandation | Norme internationale définit un ensemble de base de catégories de test de confiance et de diagnostic servant à l'utilisateur d'un système ou d'un réseau de communication pour:

- confirmer l'aptitude d'une partie spécifiée du système ou du réseau à exécuter correctement la fonction qui lui est assignée (c'est-à-dire que l'entité testée continue à fonctionner conformément à sa conception);
- effectuer des essais, à la suite de la notification ou de la détection d'un défaut, afin de déterminer la cause du problème.

Cette prescription inclut la nécessité de:

- vérifier la connexion entre deux points extrémités connus;
- vérifier que deux entités peuvent être connectées dans un délai spécifié;
- vérifier si deux entités peuvent échanger des données sans altération et mesurer le temps nécessaire à la transmission;
- vérifier que les données peuvent être envoyées et reçues sur un trajet de communication en un délai spécifié;
- déterminer si deux entités peuvent exécuter correctement des interactions protocolaires;
- vérifier le comportement observable d'une entité au niveau de ses limites;
- vérifier la capacité d'une ressource à remplir la fonction qui lui est assignée;
- vérifier la capacité d'une entité à recevoir les demandes de test entrantes et à générer les réponses appropriées.

NOTE – La présente Recommandation | Norme internationale n'a pas pour objet de définir une classification complète des types de test qui peuvent être prévus par une ressource donnée ou requis pour un usage donné.

7 Modèle

Une demande de test est envoyée à un objet, qui est géré par le conducteur de test, dont la fonction est de recevoir de telles demandes de test et d'y répondre. Comme le définit la Rec. UIT-T X.745 | ISO/CEI 10164-12, une telle fonctionnalité est appelée fonctionnalité de récepteur de demande d'action de test (fonctionnalité TARR). Les objets gérés qui font appel aux fonctionnalités soumises aux tests (objets MORT) sont identifiés explicitement ou implicitement dans les demandes de test. Chaque test met en jeu un ou plusieurs objets MORT. Dans un test, le récepteur TARR peut être un des éléments fonctionnels d'un objet MORT ou d'un autre objet géré.

Un test peut être piloté ou non piloté. Un test non piloté est un test qui n'est ni surveillé ni commandé, et dont les résultats sont présentés sous forme d'une ou plusieurs réponses à la demande de test. Un test piloté est un test pour lequel sont créés un ou plusieurs objets «tests» (TO) aux fins de commande et de surveillance. Le comportement général du test est décrit par une catégorie de test. La Figure 1 représente à titre d'exemple certaines configurations de test.

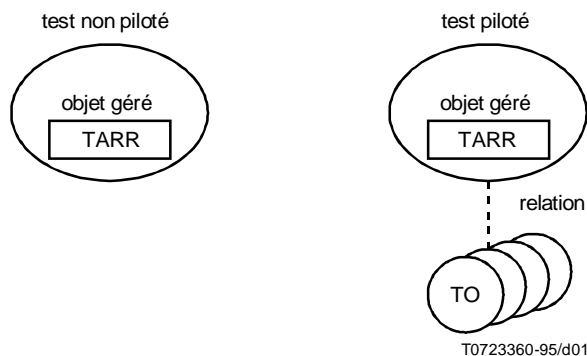


Figure 1 – Objets gérés intervenant dans tous les tests

Cet article définit les catégories de test de fiabilité et de diagnostic qui peuvent être utilisées pour étudier l'environnement OSI. Ces catégories sont les suivantes:

- test de connexion, pouvant servir à vérifier la connexion entre deux points extrémités connus;
- test de connectivité, pouvant servir à vérifier si la connexion peut être établie entre deux entités;
- test d'intégrité des données, pouvant servir à prouver que deux entités peuvent échanger des données sans altération;
- test de bouclage, pouvant servir à vérifier et à mesurer le temps nécessaire à la transmission;
- test d'intégrité de protocole, pouvant servir à examiner la capacité de deux entités de communication à exécuter correctement des interactions protocolaires;
- test de limite de ressource, dont le but est de tester le comportement d'une ressource en observant et en commandant les interactions entre la ressource et son environnement;
- autotest de ressource, pouvant être appliqué pour vérifier la capacité d'une ressource à remplir la fonction qui lui est assignée;
- test d'infrastructure d'essai, pouvant servir à étudier la capacité d'un système géré à répondre à une demande de test.

Les membres de chaque catégorie de test (par exemple, la suite de tests offerte par un système géré particulier) partagent une interface de gestion commune mais diffèrent dans certains domaines, par exemple:

- fonction ou entité physique testée;
- exhaustivité du test;
- durée du test.

Pour chaque catégorie, la présente Recommandation | Norme internationale définit:

- le but du test effectué;
- les ressources (objets MORT et AO) participant au test;
- l'environnement du test;
- le pilotage (piloté ou non piloté) du test;
- la classe d'objets TO, en cas de test piloté;
- les autres informations concernant le test;
- le mode de remise du rapport de test (sollicité ou non sollicité) et le contenu de ce rapport;
- la terminaison du test (sur achèvement normal, sur échec, sur abandon, sur demande, sur fin de délai);
- détails du minutage propre à la catégorie de test (s'il y a lieu);
- l'interface de gestion.

NOTE – La terminaison de test n'est pas équivalente à l'achèvement de test (voir la Rec. UIT-T X.745 | ISO/CEI 10164-12). La terminaison de test englobe l'achèvement du test (test réussi), l'erreur ou l'échec du test, l'abandon du test (pour les tests pilotés), l'arrêt sur fin de délai imparti et l'arrêt sur demande (pour les tests pilotés).

La présente Recommandation | Norme internationale indique également les informations nécessaires à un utilisateur pour pouvoir choisir entre les tests offerts par un système géré particulier. En général, ces informations seront spécifiées par la prescription d'une catégorie particulière de tests normalisés et/ou communiquées par le fournisseur dans le cadre de la documentation technique du système (par exemple, Recommandation V.32).

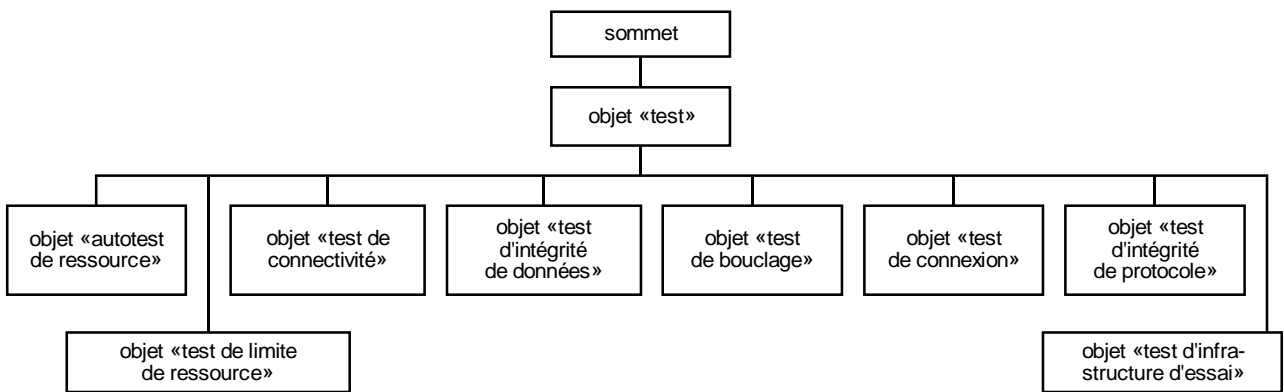
Par exemple, un commutateur téléphonique peut offrir des fonctions telles que l'acheminement et la facturation des appels mises en œuvre par des modules de matériel et de logiciel, y compris des cartes de ligne, des processeurs et des supports de mise en mémoire de données (par exemple, unités de disques). Un autotest de ressource peut être offert pour chacune des fonctions et chacun des modules.

Par le biais de la spécialisation, un objet géré peut offrir deux tests ou plus de la même fonction/du même module interne mais pour différents objectifs.

En ce qui concerne les tests pilotés, il est nécessaire de définir un objet ou des objets TO pour chaque catégorie de test. Toutes les classes d'objets TO sont issues de la classe d'objets gérés testObject définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

Ces classes d'objets TO peuvent être spécialisées afin de définir des tests plus spécifiques.

Les catégories de test ci-dessus et leurs classes d'objets «tests» associées sont définies dans ce qui suit. Les relations d'héritage entre ces classes d'objets TO sont illustrées par la Figure 2.



T0723370-95/d02

Figure 2 – Relations d'héritage entre classes d'objets «tests» de type catégorie de test

7.1 Test de connexion

7.1.1 Objet de la catégorie de test

Le test de connexion permet d'étudier l'aptitude d'un trajet de communication (réel ou virtuel) à prendre en charge un service ou un niveau de fonctionnalité désiré.

Ce test peut être mis en œuvre sous la forme d'un ou de plusieurs exercices. Les exercices exécutés sont spécifiques du type de connexion considéré et doivent être suffisants pour vérifier sa qualité de fonctionnement dans les limites documentées.

7.1.2 Prescriptions relatives aux objets MORT et AO

L'objet ou les objets MORT représentent le trajet de communication à tester. On définit deux objets associés représentant la ressource aux extrémités du trajet de communication qui émettent et reçoivent des signaux sur ce trajet de communication. Si les identités des objets MORT ou AO ne sont pas spécifiées lors du lancement du test, les défauts sont choisis par le système géré qui reçoit la demande de test. La Figure 3 illustre ces objets gérés.

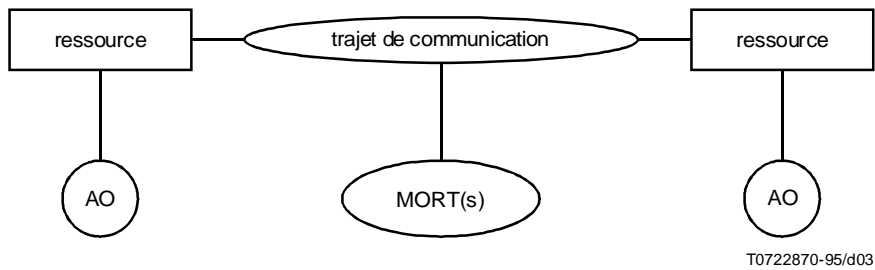


Figure 3 – Environnement de test de connexion

Dans le cas où un trajet ne prend en charge qu'un seul sens de communication entre les objets AO, l'objet AO «émetteur» applique une séquence de trames de test ou d'exercices au trajet. L'objet AO «récepteur» vérifie que les sorties correspondantes sont correctement reçues. L'ensemble exact de trames de test ou d'exercices à exécuter doit être approuvé par les objets AO qui participent au test (par exemple, il sera préprogrammé ou intégré aux deux ressources).

A noter que les diverses trames de test ou les divers exercices peuvent sensiblement différer par leur caractère et par l'aspect de la fonctionnalité du trajet qui est étudié.

Lorsqu'une ou plusieurs erreurs sont détectées, le test peut fournir, dans son compte rendu final, des informations de diagnostic susceptibles de faciliter la localisation et l'élimination des dérangements (par exemple, «rupture de câble suspectée»).

Lorsqu'un trajet assure une communication bidirectionnelle selon l'indication de la direction de test d'une demande de test simple, des opérations du type décrit ci-dessus sont exécutées séparément pour chaque sens de communication. L'ordre selon lequel les sens de communication seront testés dépend de l'application. La Figure 4 montre un exemple de transmission de trames de test.

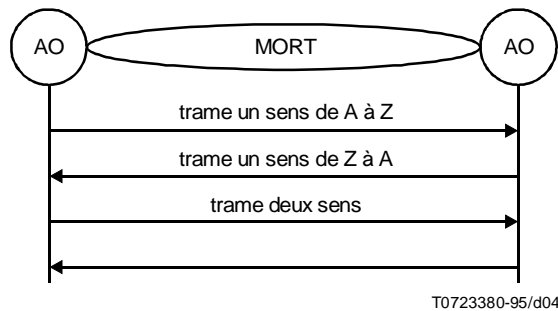


Figure 4 – Exemple d'utilisation de la catégorie de test de connexion

7.1.3 Environnement de test

Le test de la connexion peut être de type intrusif ou non à la fois par rapport aux objets MORT et aux objets associés (AO).

S'il est de type intrusif, il sera abandonné dans les conditions suivantes:

- l'objet ou les objets MORT ou AO n'ont pas été préalablement placés dans un état approprié (par exemple, réservés pour le test);
- l'objet ou les objets MORT ou AO ne peuvent être placés dans un état approprié lors du lancement du test.

7.1.4 Test piloté/non piloté

Le test peut être modélisé sous la forme d'un test piloté ou non piloté.

7.1.5 Prescriptions d'objets «tests»

La présente Recommandation | Norme internationale définit un objet «test de connexion». Cet objet «test» peut être spécialisé afin de définir des classes d'objets «tests» plus spécifiques.

7.1.6 Lancement spécifique de la catégorie de test

Lorsqu'un test non piloté est demandé, il convient de spécifier la période de temporisation définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres optionnels suivants, définis dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 sont également nécessaires:

- identité des objets associés et, le cas échéant, informations complémentaires relatives aux objets AO;
- identité de l'objet MORT;
- temporisation d'exécution.

Les paramètres optionnels suivants peuvent s'avérer nécessaires:

- trame de test à utiliser au cours du test. En l'absence de ce paramètre, la trame est propre à l'instance;
- direction du test appliqué à l'objet MORT. En l'absence de ce paramètre, l'objet MORT représente la direction de transmission à l'extrémité locale d'un trajet unidirectionnel;
- durée affectée au test. En l'absence de ce paramètre, la durée du test est propre à l'instance;
- seuil utilisé comme critère pour déterminer le résultat du test. En l'absence de ce paramètre, le seuil est propre à l'instance;
- espacement des comptes rendus des résultats de test intermédiaires. En l'absence de ce paramètre, les comptes rendus intermédiaires n'existent pas ou leur espacement est propre à l'instance.

7.1.7 Événements déclenchant le compte rendu du résultat et la fin du test

Les événements qui déclenchent un compte rendu implicite des résultats intermédiaires sont les suivants:

- achèvement des opérations de test effectuées pour un seul sens de communication;
- expiration du délai spécifié de compte rendu.

Les événements qui déclenchent un compte rendu implicite des résultats sont les suivants:

- achèvement de l'exercice ou des exercices du test effectués sur un seul sens de communication;
- achèvement du test;
- expiration de la période de temporisation;
- si le test est modélisé sous la forme d'un test piloté, réception d'une demande de fin du test.

Les événements qui déclenchent la fin du test sont les suivants:

- achèvement du test;
- échec de l'exécution normale du test;
- expiration du délai imparti pour l'achèvement du test (fin de temporisation);
- si le test obéit à un modèle piloté, réception d'une demande de terminaison.

7.1.8 Compte rendu du résultat

Lorsque le test obéit à un modèle non piloté, le compte rendu implicite sera utilisé et viendra en réponse à une ou plusieurs actions. Lorsque le test obéit à un modèle piloté, le compte rendu sollicité sera pris en charge et sera déclenché par une demande de lecture; le compte rendu conditionnellement implicite peut également être utilisé au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres optionnels suivants sont également nécessaires dans le compte rendu du résultat:

- identité de l'objet ou des objets MORT si elle n'est pas spécifiée lors du lancement du test;
- identités des objets AO si elles ne sont pas spécifiées lors du lancement du test;
- s'il s'agit du compte rendu final, résultat du test choisi comme indiqué ci-après.

Les paramètres optionnels suivants peuvent être également retournés:

- signal de test reçu;
- taux d'erreur détecté au cours du test;
- direction du test appliqué à l'objet MORT;
- durée de test;
- informations détaillées sur le résultat du test (en fonction de la mise en œuvre);
- si le résultat est un échec, informations complémentaires de diagnostic pour faciliter la localisation et l'élimination des dérangements;
- autres informations de résultat en fonction de l'exécution du test.

Le résultat du test est déterminé comme suit:

- si tous les exercices ont été exécutés correctement, le résultat indiqué sera «PASS» (succès);
- si un exercice quelconque a détecté une erreur, le résultat indiqué sera «FAIL» (échec);
- si la durée du test a dépassé la période de temporisation spécifiée, le résultat indiqué sera «TIMED OUT» (expiration);
- si le test a pris fin à cause d'une erreur de procédure ou de la réception d'une primitive TERMINATION REQUEST (demande de fin de test), le résultat indiqué sera «PREMATURE TERMINATION» (fin de test prématurée);
- dans le cas contraire, le résultat indiqué sera «INCONCLUSIVE» (non concluant).

7.1.9 Fin du test

Lorsque le test n'est pas piloté, il utilise l'arrêt implicite.

Lorsque le test est piloté, on utilise aussi bien l'arrêt implicite qu'explicite. Pour l'arrêt explicite, on utilise le service TEST-TERMINATION (arrêt de test) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

7.2 Test de connectivité

7.2.1 Objet de la catégorie de test

Le test de connectivité sert à vérifier que la connectivité peut être établie entre deux entités (représentées par un objet MORT et par un objet associé) dans un laps de temps spécifié.

7.2.2 Prescriptions relatives à l'objet MORT et à l'objet associé

L'appel de ce test met en jeu deux entités représentées par deux objets gérés (l'objet MORT et l'objet associé AO) entre lesquels la connectivité est à vérifier. Une fois à l'état de test, l'objet MORT tente d'établir le contact avec l'objet associé. Cela se traduira par exemple, dans le cas d'un protocole en mode connexion, par l'établissement d'une connexion, ou dans le cas d'un protocole en mode sans connexion, par un échange d'unités de données (identificateurs XID par la commande LLC par exemple). On mesure le temps qui s'écoule entre l'envoi de la demande par l'objet MORT et la réception de la confirmation correspondante par l'objet associé, qui correspondra au temps d'établissement d'appel. Une temporisation peut être associée au test. Cette temporisation peut être spécifiée indirectement sous la forme d'une valeur d'attribut de temporisateur fournie par l'objet MORT. Les Figures 5 et 6 illustrent l'utilisation des tests de connectivité.

7.2.3 Environnement de test

Le test de connectivité est intrusif par rapport à la fois à l'objet MORT et à l'objet AO. Il sera abandonné dans les conditions suivantes:

- si l'objet MORT ou l'objet AO n'ont pas été préalablement placés dans un état approprié (réservation pour test par exemple);
- si l'objet MORT ou l'objet AO ne peuvent être placés dans un état approprié lors du lancement du test.

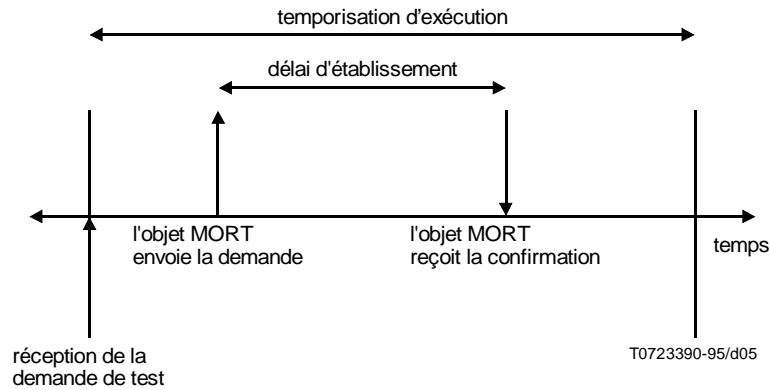


Figure 5 – Test de connectivité – exemple 1

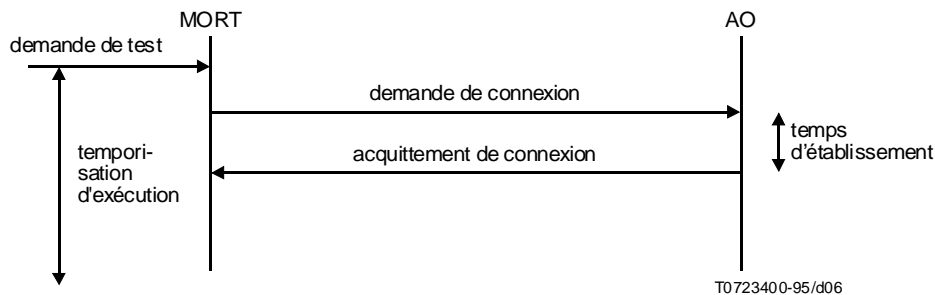


Figure 6 – Test de connectivité – exemple 2

7.2.4 Tests pilotés/non pilotés

Un test de connectivité peut être modélisé sous la forme d'un test piloté ou non piloté.

7.2.5 Prescriptions pour l'objet «test»

La présente Recommandation | Norme internationale définit un objet «test de connectivité». Cet objet «test» peut être spécialisé pour définir des classes d'objets «tests» plus spécifiques.

7.2.6 Lancement spécifique de la catégorie de test

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres optionnels suivants définis dans la même Recommandation sont également nécessaires:

- identité de l'objet associé et, le cas échéant, informations complémentaires s'y rapportant;
- identité de l'objet MORT s'il n'est pas implicitement connu du récepteur TARR;
- temporisation d'exécution, s'il n'est pas implicitement connu de l'objet MORT.

7.2.7 Événements déclenchant le compte rendu du résultat et la fin du test

Les événements qui déclenchent la notification implicite des résultats et la fin implicite du test sont les suivants:

- réception d'une confirmation émise par l'objet associé avant l'expiration de la temporisation d'exécution;
- expiration de la temporisation d'exécution.

7.2.8 Compte rendu du résultat du test

Lorsque le test obéit à un modèle non piloté, le compte rendu implicite sera utilisé et viendra en réponse à une ou plusieurs actions. Lorsque le test obéit à un modèle piloté, le compte rendu sollicité sera pris en charge et sera déclenché par une demande de lecture; le compte rendu conditionnellement implicite peut également être utilisé au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres suivants sont également nécessaires:

- identité de l'objet associé si elle n'a pas été explicitement spécifiée dans la demande de test;
- identité de l'objet MORT si elle n'a pas été explicitement spécifiée dans la demande de test.

Les paramètres optionnels suivants peuvent être également renvoyés:

- s'il s'agit du rapport final, résultat du test choisi selon le cas en fonction de la Rec. UIT-T X.745 | ISO/CEI 10164-12;
- si le résultat indiqué est «PASS» (succès), durée d'établissement en unités de temps;
- si le résultat indiqué est «FAIL» (échec), temps écoulé entre l'émission de la demande par l'objet MORT et la réception de la confirmation correspondante de l'objet AO ou la fin de la demande;
- toute information complémentaire spécifique du test.

La valeur du résultat du test à renvoyer dans le compte rendu final est déterminée comme suit:

- si une confirmation positive est reçue avant la période d'expiration de la temporisation, le résultat indiqué sera «PASS» (succès);
- si une confirmation négative est reçue, le résultat sera «FAIL» (échec);
- si la confirmation n'a pas été reçue avant l'expiration d'une période de temps préalablement spécifiée, le résultat indiqué sera «Timed-Out» (expiration de la temporisation);
- si le test a pris fin à cause d'une erreur de procédure ou de la réception d'une primitive TERMINATE REQUEST (demande de fin de test), le résultat indiqué sera «PREMATURE TERMINATION» (fin de test prématurée);
- dans le cas contraire, le résultat indiqué sera «INCONCLUSIVE» (non concluant).

7.2.9 Fin de test

En mode non piloté, cette catégorie de test utilise l'arrêt implicite.

En mode piloté, on utilise aussi bien l'arrêt implicite qu'explicite. Pour l'arrêt explicite, on utilise le service de fin de test (TEST-TERMINATION) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

7.3 Test d'intégrité des données

7.3.1 Objet de la catégorie de test

Le test d'intégrité des données sert à vérifier si deux entités peuvent échanger des données sans aucune altération et à mesurer le temps nécessaire à un tel échange.

7.3.2 Prescriptions relatives à l'objet MORT et à l'objet associé

Pendant qu'il est à l'état de test, l'objet MORT transmet des données à un objet associé. Celui-ci, dès qu'il les reçoit, renvoie à l'objet MORT une copie des données. Le transport des données peut s'effectuer en mode connexion ou en mode sans connexion. En mode connexion, l'objet MORT doit s'assurer qu'il existe une connexion préalablement à l'envoi des données. Pendant l'établissement de la connexion, l'objet «test» est placé dans l'état d'initiation de test. Pendant la libération de la connexion, l'objet «test» passe à l'état de fin de test. Les Figures 7 et 8 illustrent l'utilisation du test d'intégrité des données.

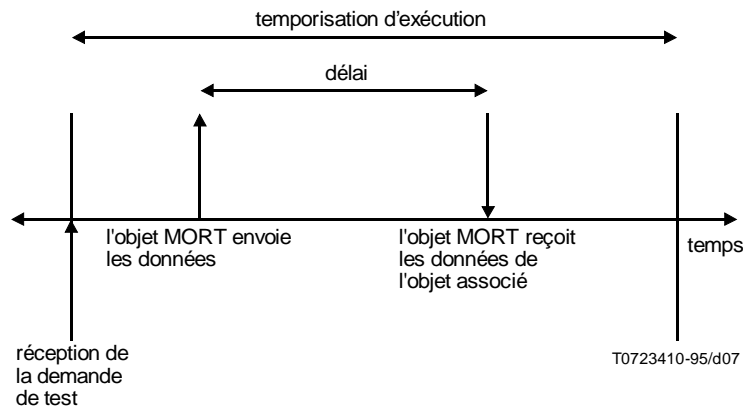


Figure 7 – Test d'intégrité des données – exemple 1

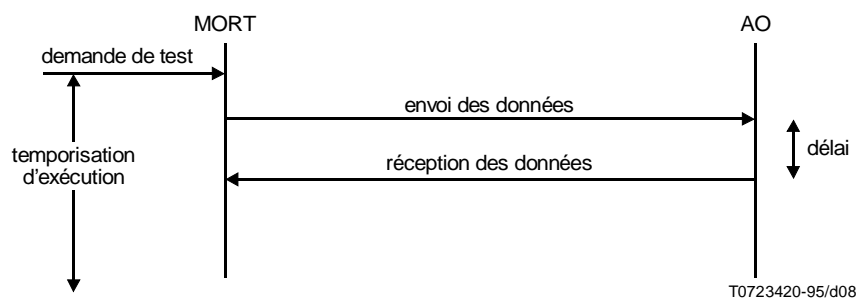


Figure 8 – Test d'intégrité des données – exemple 2

7.3.3 Environnement de test

Le test d'intégrité des données est intrusif par rapport à la fois à l'objet MORT et à l'objet AO. Il sera abandonné dans les conditions suivantes:

- si l'objet MORT ou l'objet AO n'ont pas été préalablement placés dans un état approprié (réservation pour test par exemple);
- si l'objet MORT ou l'objet AO ne peuvent être placés dans un état approprié lors du lancement du test.

7.3.4 Tests pilotés/non pilotés

Le test d'intégrité des données peut être modélisé sous la forme d'un test piloté ou non piloté.

7.3.5 Classe d'objets TO

La présente Recommandation | Norme internationale définit un objet «test d'intégrité de données». Cet objet «test» peut être spécialisé pour définir des classes d'objets «tests» plus spécifiques.

7.3.6 Lancement spécifique de la catégorie de test

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres optionnels suivants définis dans la même Recommandation sont également nécessaires:

- identité de l'objet associé et, le cas échéant, informations complémentaires s'y rapportant;
- identité de l'objet MORT s'il n'est pas implicitement connu du récepteur TARR;
- temporisation d'exécution, s'il n'est pas implicitement connu de l'objet MORT.

Le paramètre optionnel suivant peut être spécifié:

- unités de données utilisées dans le test. Si elles ne sont pas indiquées, les unités de test sont propres à l'objet.

7.3.7 Événements déclenchant la notification du résultat et l'arrêt du test

Les événements qui déclenchent la notification implicite des résultats et la fin implicite du test sont les suivants:

- réception d'une confirmation émise par l'objet associé avant l'expiration de la temporisation d'exécution;
- expiration de temporisation d'exécution.

7.3.8 Compte rendu du résultat du test

Lorsque le test obéit à un modèle non piloté, le compte rendu implicite sera utilisé et viendra en réponse à une ou plusieurs actions. Lorsque le test obéit à un modèle piloté, le compte rendu sollicité sera pris en charge et sera déclenché par une demande de lecture; le compte rendu conditionnellement implicite peut également être utilisé au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres suivants sont également nécessaires:

- identité de l'objet associé si elle n'a pas été explicitement spécifiée dans la demande de test;
- identité de l'objet MORT si elle n'a pas été explicitement spécifiée dans la demande de test.

Les paramètres optionnels suivants peuvent être également renvoyés:

- s'il s'agit du rapport final: résultat du test choisi selon le cas en fonction de la Rec. UIT-T X.745 | ISO/CEI 10164-12;
- si le résultat indiqué est «PASS» (succès): délai effectif jusqu'au retour des données;
- si le résultat indiqué est «FAIL» (échec): temps écoulé entre l'émission de la demande par l'objet MORT et la réception de la confirmation correspondante de l'objet AO ou la fin de la demande;
- si le test a échoué à cause des données reçues: les données reçues. En outre, si les unités de données utilisées dans le test n'ont pas été fournies dans le paramètre de demande de test, les unités de données doivent être également renvoyées;
- toute information complémentaire propre au test.

Le résultat du test est renvoyé dans le compte rendu final. La valeur du résultat du test est déterminée comme suit:

- si toutes les unités de données ont été transmises et reçues avec succès, le résultat du test est «PASS» (succès);
- si des erreurs de transmission ont été détectées, le résultat du test est «FAIL» (échec);
- si la durée du test dépasse un délai préalablement spécifié, le résultat sera «TIMEOUT» (fin de temporisation);
- si le test a pris fin à cause d'un échec de procédure ou de la réception d'une primitive TERMINATE REQUEST (demande de fin de test), le résultat est «PREMATURE TERMINATION» (fin de test prématurée);
- dans les autres cas, le résultat est «INCONCLUSIVE» (pas concluant).

7.3.9 Fin de test

En mode non piloté, cette catégorie de test utilise l'arrêt de test implicite.

En mode piloté, on utilise aussi bien l'arrêt implicite qu'explicite. Pour l'arrêt explicite, on utilise le service de fin de test (TEST-TERMINATION) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

7.4 Test de bouclage

7.4.1 Objet de la catégorie de test

Le test de bouclage sert à vérifier que des données peuvent être envoyées et reçues sur une voie de communication dans un temps aller-retour spécifié et avec un taux d'erreurs acceptable. Le bouclage peut être réalisé de plusieurs manières, par exemple par un bouclage matériel ou par le renvoi des données en écho. Les bouclages peuvent être analogiques ou numériques. Il peut exister des bouclages matériels non transparents, des bouclages matériels transparents, des bouclages matériels de charge utile, des renvois de données en écho, etc. Le test de bouclage peut être effectué sur une voie de communication en mode connexion ou sans connexion.

7.4.2 Prescriptions relatives à l'objet MORT et à l'objet associé

7.4.2.1 Configuration de bouclage

Pour pouvoir invoquer les tests de bouclage, il faut spécifier la configuration dans laquelle placer les objets MORT et AO. Les objets MORT représentent les ressources soumises au test de bouclage. Ils n'englobent cependant pas forcément la totalité du chemin de communication sur lequel pourront exister d'autres objets gérés non identifiés à tester. Les objets associés représentent les ressources assurant le bouclage. La spécification et l'utilisation de ces objets associés dépendent de la réalisation. Si les objets associés ne sont pas spécifiés, on supposera que la configuration fait intervenir un bouclage manuel.

Dans certains cas, il est souhaitable de pouvoir spécifier qu'un bouclage intervient dans le cadre d'un objet MORT. Ce peut être le cas lorsqu'une ressource est modélisée sous la forme d'un objet géré isolé tout en étant en fait assez complexe pour que des bouclages puissent se produire à plusieurs endroits dans le cadre de l'objet MORT (par exemple un «trajet» ou «conduit» peut être modélisé sous la forme d'un objet MO isolé tout en étant en fait composé de plusieurs sections dont chacune peut être rebouclée). Pour que cette situation puisse être modélisée, il est permis que des objets AO soient présents à des endroits où ils ne testeront qu'une partie de l'objet MORT.

Les Figures 9 et 10 montrent deux configurations possibles d'objets MORT et AO.

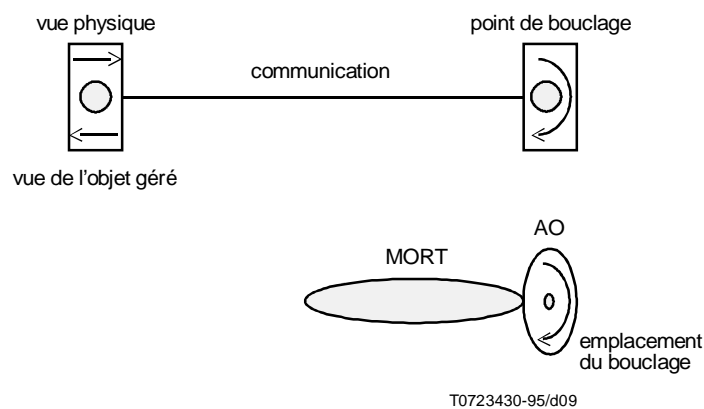


Figure 9 – Configuration de bouclage – Premier cas

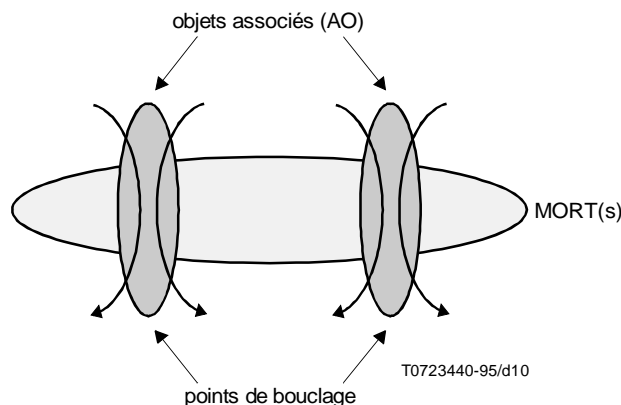


Figure 10 – Configuration de bouclage – Deuxième cas

La première configuration montre un exemple de bouclage simple. L'objet MORT est un objet qui modélise le chemin de communication entre les deux ressources. Un objet associé unique modélise la ressource qui assure le bouclage. Le positionnement du bouclage dans l'objet est indiqué. La ressource comportant le point à partir duquel les données de test sont injectées n'est pas modélisée dans cet exemple.

La deuxième configuration représente la capacité d'invoquer un bouclage à différents endroits pour le même objet MORT. Dans ce cas, le gestionnaire spécifiera l'objet associé à utiliser pour le test de bouclage.

Le gestionnaire peut demander l'exécution de plusieurs tests de bouclage sur le même objet ou ensemble d'objets MORT, en spécifiant plusieurs objets associés. Les tests sont dans ce cas appliqués dans un ordre non spécifié, à partir de l'objet «test» disposant de la fonctionnalité de récepteur TARR vers chaque objet associé.

7.4.2.2 Terminologie de mesure des temps dans le test

En plus de la temporisation d'exécution utilisée optionnellement par toutes les catégories de test de la présente Recommandation | Norme internationale, un certain nombre de paramètres temporels servent à contrôler le déroulement du test de bouclage. Le présent paragraphe définit différents termes de mesure de temps, illustrés Figures 11 et 12, et fournit des exemples sur la manière pour le conducteur de test de spécifier un test de bouclage correct et utile.

La Figure 11 représente un test comportant l'envoi de trois trames de test distinctes, les résultats étant notifiés à la fin de chaque trame. La Figure 12 représente un test comportant l'envoi d'une seule trame longue, des résultats étant notifiés à des instants intermédiaires. De nombreuses autres combinaisons peuvent être conçues à partir de ces concepts.

Les termes servant à décrire les différents aspects de mesure des temps sont les suivants:

- *instant de départ* – Sert à programmer le test pour un instant futur donné. La temporisation d'exécution est décomptée à partir de cet instant. (L'*instant de fin* n'est pas représenté, car la présence simultanée d'un instant de fin et d'une temporisation d'exécution est à la fois redondant et source d'erreur, les événements correspondants provoquant tous deux l'indication «fin de temporisation» dans le paramètre résultat du compte rendu de test);
- *durée de trame* – Durée de transmission de chaque trame. Ce temps peut être exprimé directement en secondes, ou en termes de bits, d'octets, et (optionnellement) de paquets ou de blocs à un débit donné;
- *périodicité de notification* – S'il n'est pas nul ('0' signifiant pas de compte rendu intermédiaire), intervalle de temps séparant la notification des résultats intermédiaires. Comme le montre la Figure 12, un compte rendu intermédiaire sera envoyé à intervalles d'une période de notification, à compter de l'instant égal à l'instant de départ plus une période;
- *temps de propagation aller-retour* – Délai correspondant à la transmission d'une trame jusqu'au point de bouclage et retour. Il s'agit d'une mesure physique du système sous test;
- *temporisation aller-retour* – Spécifie le temps durant lequel l'exécutant du test attendra le retour des données. Elle est mesurée depuis l'instant d'émission de la trame jusqu'à la réception des données renvoyées par le circuit bouclé. Si le temps de propagation aller-retour est supérieur à la valeur spécifiée, l'ensemble du test est considéré comme ayant échoué et un résultat d'échec est retourné;
- *espacement des tests* – Délai séparant la fin de la réception d'une trame de test et le début d'émission de la suivante. Par souci de clarté, l'espacement représenté Figure 11 est assez important, mais dans la plupart des cas, la succession sera quasiment instantanée, c'est-à-dire que l'émission d'une trame commencera aussitôt la fin de réception de la trame précédente;
- *temporisation d'exécution* – Délai imparti à l'achèvement de l'ensemble du test. Dans un test de bouclage, ce délai doit être supérieur au nombre de trames multiplié par la durée de trame plus le temps de propagation aller-retour. Il est possible d'utiliser un instant d'arrêt aux mêmes fins.

7.4.2.3 Déroulement du test

La demande de test spécifie l'objet MORT (ou l'ensemble d'objets MORT) constituant tout ou partie du chemin de communication du test. Si la configuration multiple de l'objet MORT n'est pas mise en évidence par les informations de configuration statique normales (par dénomination ou relation), la demande de test ne devra pas faire référence à de multiples objets MORT.

L'identificateur de la demande de test indique qu'un test de bouclage est en demande; il spécifie également des paramètres de bouclage spécifiés, tels que le type de données pour le test, l'instant de début du test, la temporisation de compte rendu, le seuil d'erreur et la durée du test. On peut spécifier un paramètre de temps d'établissement

- a) d'environnement de test;
- b) d'armement (automatique ou manuel) du bouclage.

Le test sera exécuté à l'expiration du temps d'établissement.

La non-réception des données dans le délai imparti par la temporisation aller-retour, ou la réception des données entachées d'un taux d'erreur excessif, entraînera un diagnostic d'échec (FAIL). A la fin du test, un diagnostic de succès (PASS) sera donné si toutes les données ont été émises et reçues et que le taux d'erreur sur les unités de données ne dépasse pas le paramètre de seuil d'erreur.

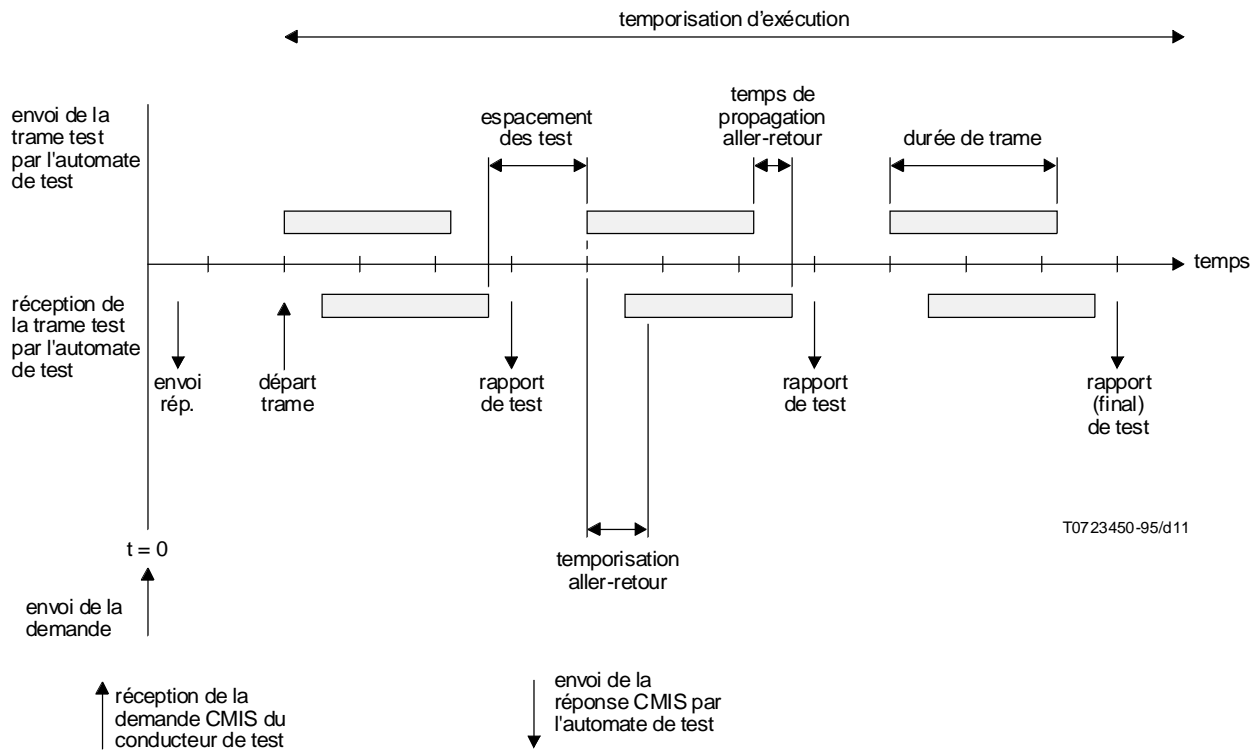


Figure 11 – Terminologie de mesure des temps dans un test multiframe

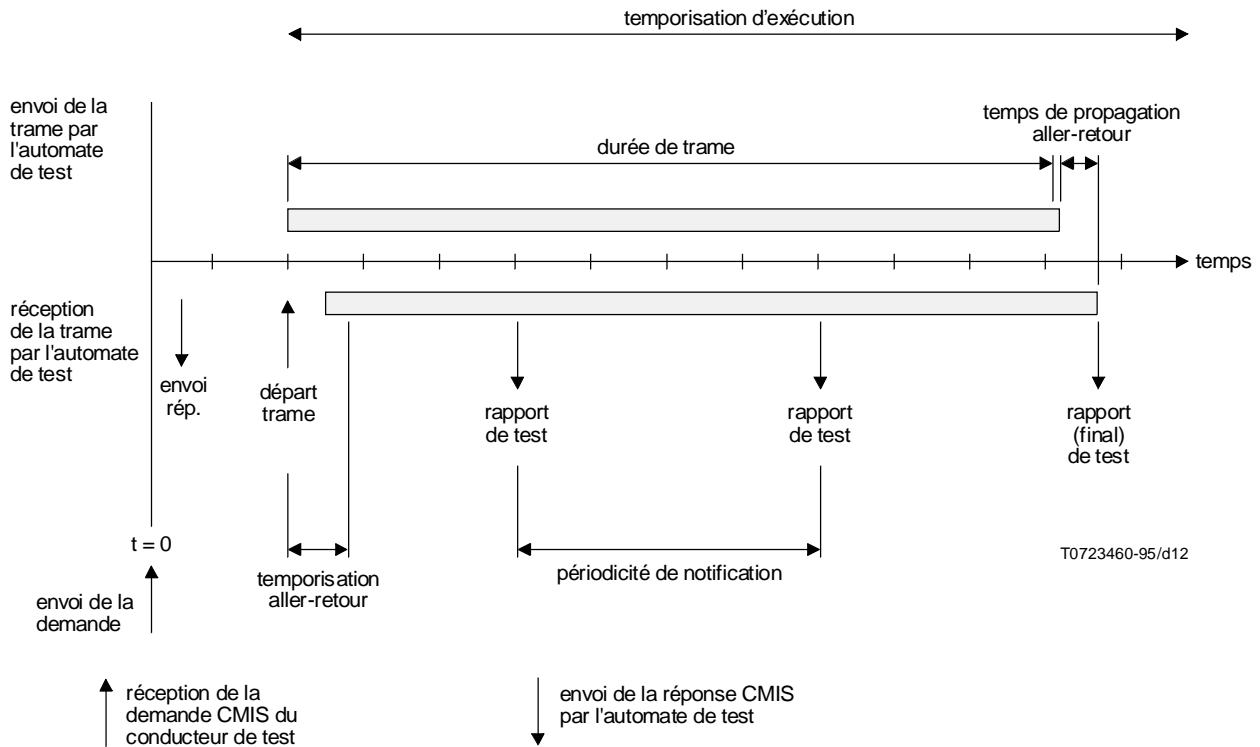


Figure 12 – Terminologie de mesure des temps dans un test monoframe

Dans les tests pilotés, l'objet possédant la fonctionnalité de récepteur TARR peut créer un ou plusieurs objets «tests» TO. Si une même demande de test spécifie plusieurs points de bouclage, un objet test sera utilisé pour chaque test de bouclage.

Le test sera exécuté pendant la durée spécifiée par le paramètre temporisation d'exécution préalablement choisi ou véhiculé par le champ d'information de la demande de test. Si plusieurs bouclages sont demandés, le paramètre d'information additionnelle de la demande de test indiquera les multiples temporisations aller-retour et taux d'erreur.

Pendant le test, le conducteur du test peut obtenir des résultats intermédiaires. Le résultat du test est envoyé dans le compte rendu final de résultat de test.

7.4.3 Environnement de test

Le test d'intégrité des données peut être intrusif ou non par rapport à l'objet MORT et à l'objet AO.

Le test ne peut avoir lieu s'il est impossible de verrouiller l'objet MORT. Celui-ci sera déverrouillé dès l'arrêt (normal ou anormal) du test.

Si le test est intrusif, il sera abandonné dans les conditions suivantes:

- si l'objet MORT ou l'objet AO n'ont pas été préalablement placés dans un état approprié (réservation pour test par exemple);
- si l'objet MORT ou l'objet AO ne peuvent être placés dans un état approprié lors du lancement du test.

7.4.4 Tests pilotés/non pilotés

Cette catégorie de test peut être modélisée par un test piloté ou non piloté.

7.4.5 Prescriptions d'objet «test»

La présente Recommandation | Norme internationale définit un objet «test de bouclage». Cet objet «test» TO peut être spécialisé pour définir des classes d'objets TO plus spécifiques.

7.4.6 Lancement spécifique de la catégorie de test

Les paramètres suivants, qui sont optionnels dans l'ISO/CEI 10164-12, sont requis pour le lancement du test de bouclage:

- identité de l'objet associé s'il ne s'agit pas d'un bouclage manuel et, le cas échéant, informations complémentaires s'y rapportant (point de bouclage dans l'objet associé par exemple);
- identité de l'objet MORT si elle n'est pas implicitement connue du récepteur TARR;
- temporisation d'exécution, si elle n'est pas établie par accord préalable ou si le test n'est pas piloté.

Les paramètres suivants peuvent également s'avérer nécessaires:

- trames de données utilisées pour le test. En l'absence de ce paramètre, la trame de test sera propre à la réalisation;
- instant de départ du test;
- espacement des trames transmises;
- périodicité de notification des résultats de test intermédiaires;
- type de bouclage (bouclage physique, par renvoi en écho ou les deux). En l'absence de ce paramètre, le type de bouclage est propre à la réalisation;
- temporisation aller-retour pour le test;
- seuil d'erreur en bouclage.

7.4.7 Événements déclenchant une notification de résultat et l'arrêt de test

Les événements déclenchant la notification implicite des résultats sont les suivants:

- réception de toutes les données émises avant l'expiration de la durée du test;
- expiration d'une temporisation aller-retour;
- expiration de la temporisation d'exécution du test, qui englobe le temps d'initiation et les temporisations aller-retour;
- instant déterminé par la périodicité de notification;
- réception d'une demande d'arrêt de test en mode piloté.

Les événements qui déclenchent l'arrêt implicite du test sont les suivants:

- achèvement du test;
- transgression d'un seuil d'erreur;
- dépassement d'une temporisation aller-retour;
- dépassement de la temporisation d'exécution du test.

7.4.8 Compte rendu du résultat du test

Lorsque le test obéit à un modèle non piloté, le compte rendu implicite sera utilisé et viendra en réponse à une ou plusieurs actions. Lorsque le test obéit à un modèle piloté, le compte rendu sollicité sera pris en charge et sera déclenché par une demande de lecture; le compte rendu conditionnellement implicite peut également être utilisé au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres suivants sont également nécessaires:

- identité des objets associés;
- en cas de succès du test, les données envoyées en retour et le taux d'erreur;
- en cas d'échec du test, la cause de l'échec (dépassement du seuil d'erreur ou de la temporisation aller-retour);
- le résultat de test choisi selon le cas en fonction de la Rec. UIT-T X.745 | ISO/CEI 10164-12 doit être renvoyé s'il s'agit du compte rendu final.

Les paramètres optionnels suivants peuvent être également renvoyés:

- identité des objets MORT si elle n'est pas explicitement spécifiée dans la demande de test;
- toute information complémentaire propre au test.

La valeur du résultat du test est déterminée comme suit:

- si toutes les trames de test ont été transmises et renvoyées avec succès et que le seuil d'erreur n'a pas été dépassé, le résultat indiqué est «PASS» (succès);
- si le seuil d'erreur ou la temporisation aller-retour sont dépassés pendant un des tests de bouclage, le résultat indiqué est «FAIL» (échec);
- si la durée du test dépasse la temporisation préalablement spécifiée, le résultat du test est «TIMEOUT» (expiration de la temporisation);
- si le test prend fin à cause d'une erreur de procédure ou à la suite de la réception d'une primitive TERMINATE REQUEST (demande de fin de test), le résultat indiqué est «PREMATURE TERMINATION» (fin de test prématurée);
- dans les autres cas, le résultat indiqué est «INCONCLUSIVE» (non concluant).

7.4.9 Fin de test

En mode non piloté, cette catégorie de test utilise l'arrêt de test implicite.

En mode piloté, on utilise aussi bien l'arrêt implicite qu'explicite. Pour l'arrêt explicite, on utilise le service de fin de test (TEST-TERMINATION) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

7.5 Test d'intégrité de protocole

7.5.1 Objet de la catégorie de test

L'objet du test d'intégrité de protocole est de vérifier si l'objet MORT peut mener correctement des interactions protocolaires avec un objet associé spécifié.

Ce test n'a pas pour fonction de consigner les éventuels échanges protocolaires résultants; une telle fonction peut être assumée par un ou plusieurs registres qui consigneraient les notifications correspondant à l'émission et à la réception des éléments de protocole.

7.5.2 Prescriptions relatives à l'objet MORT et à l'objet associé

Aucun échange préalable n'est nécessaire entre l'objet MORT et l'objet associé avant l'échange des éléments de protocole du test. L'objet MORT émet des éléments de protocole selon une séquence spécifiée. Un élément de protocole donné peut être transmis à une des conditions suivantes:

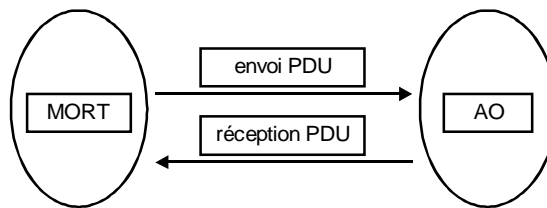
- 1) instant de départ du test (le premier élément de protocole seulement);
- 2) écoulement d'un intervalle de temps donné depuis la dernière transmission;
- 3) réception de l'élément de protocole correct en réponse dans le délai imparti.

L'objet ou les objets associés manifesteront simplement un comportement normal en réponse aux éléments de protocole reçus.

Le test se termine implicitement lorsque la séquence complète d'éléments de protocole a été transmise ou qu'une erreur de séquencement s'est produite dans la série d'éléments de protocole ou, explicitement, lorsque le conducteur du test donne l'ordre de terminer le test.

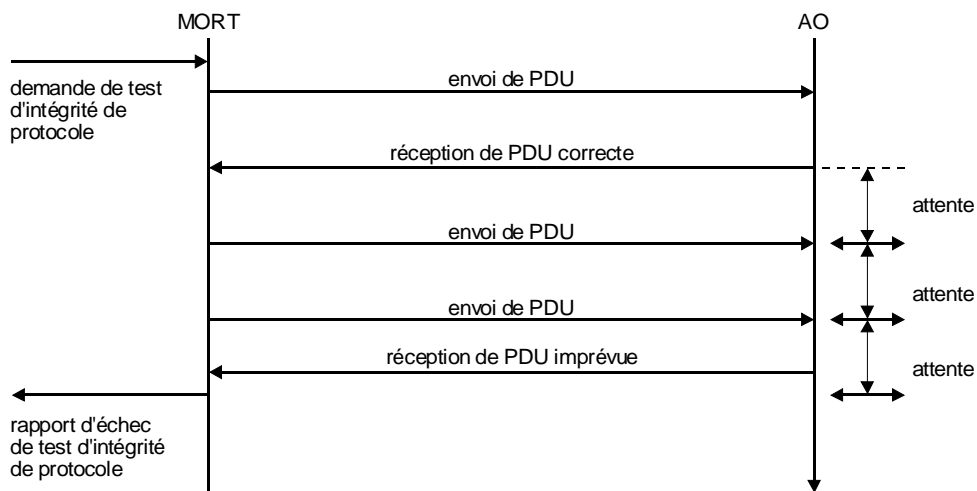
La notification du résultat est effectuée implicitement à la fin du test. Le résultat du test est renvoyé dans le compte rendu final de résultat de test.

La Figure 13 représente les objets gérés intervenant dans le test d'intégrité de protocole, et la Figure 14 illustre un test d'intégrité aboutissant à un diagnostic d'échec.



T0723470-95/d13

Figure 13 – Environnement du test d'intégrité de protocole



T0723480-95/d14

Figure 14 – Exemple de test d'intégrité de protocole

7.5.3 Environnement de test

Le test d'intégrité de protocole est généralement non intrusif par rapport aux objets MORT et aux objets associés. Toutefois, il sera abandonné dans les conditions suivantes:

- si l'objet MORT ou l'objet AO n'ont pas été préalablement placés dans un état approprié (réservation pour test par exemple);
- si l'objet MORT ou l'objet AO ne peuvent être placés dans un état approprié lors du lancement du test.

7.5.4 Tests pilotés/non pilotés

Cette catégorie de test peut être modélisée sous la forme d'un test piloté ou non piloté.

7.5.5 Prescriptions d'objets «tests» TO

La présente Recommandation | Norme internationale définit un objet «test d'intégrité de protocole». Cet objet «test» peut être spécialisé pour définir des classes d'objets «tests» plus spécifiques.

7.5.6 Initiation propre à la catégorie de test

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres optionnels suivants définis dans la même Recommandation sont également nécessaires:

- identité de l'objet associé et, le cas échéant, informations complémentaires s'y rapportant;
- identité de l'objet MORT s'il n'est pas implicitement connu du récepteur TARR.

Les paramètres optionnels suivants peuvent être également spécifiés:

- périodicité de notification de résultat de test si on désire la notification de résultats de test intermédiaires;
- unité PDU à transmettre;
- unité PDU attendue en réponse (facultatif);
- condition à remplir avant l'envoi de l'unité PDU suivante. Il peut s'agir d'un intervalle de temps fixe ou de la réception de l'unité PDU prévue avant un intervalle d'attente;
- délai à interpréter comme étant le délai entre la transmission des unités PDU (si la condition est un intervalle de temps fixe) ou comme étant la période de temporisation (lorsque la poursuite de la procédure à la réception de l'unité PDU est choisie).

Lorsqu'un test d'intégrité de protocole non piloté est demandé, le paramètre de période de temporisation défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 doit être spécifié.

7.5.7 Événements déclenchant la notification du résultat et la fin du test

Les événements qui déclenchent le compte rendu du résultat sont les suivants:

- réalisation des conditions de fin du test.

Les événements qui déclenchent la fin du test sont les suivants:

- réception de la demande de fin de test s'il s'agit d'un test piloté;
- achèvement du dernier élément de protocole de la séquence;
- non-réalisation des conditions de passage à l'unité PDU suivante de la séquence.

7.5.8 Compte rendu du résultat du test

Lorsque le test obéit à un modèle non piloté, le compte rendu implicite sera utilisé et viendra en réponse à une ou plusieurs actions. Lorsque le test obéit à un modèle piloté, le compte rendu sollicité sera pris en charge et sera déclenché par une demande de lecture; le compte rendu conditionnellement implicite peut également être utilisé au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

Des comptes rendus de résultat intermédiaires sont envoyés après chaque élément de protocole testé. Le compte rendu final de résultat de test est envoyé à la fin du test.

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres suivants sont nécessaires:

- identité des objets associés;
- identité des objets MORT, si elle n'est pas explicitement spécifiée dans la demande de test.

Les paramètres optionnels suivants peuvent être également renvoyés:

- s'il s'agit du compte rendu final, le résultat du test choisi selon le cas en fonction de la Rec. UIT-T X.745 | ISO/CEI 10164-12;
- toute information complémentaire spécifique du test;
- en cas de succès:
 - temps écoulé entre le lancement du test et l'achèvement du test;
- en cas d'échec:
 - pour chaque unité PDU attendue, si elle a été reçue avec succès, si elle n'a pas été renvoyée ou si une unité PDU incorrecte a été reçue;
 - si une unité PDU incorrecte est reçue, l'unité PDU effectivement reçue.

La valeur du résultat du test est déterminée comme suit:

- si les éléments de protocole ont été transmis et reçus avec succès, le résultat du test est «PASS» (succès);
- si des erreurs de transmission ou de réception d'éléments de protocole ont été détectées, le résultat est «FAIL» (échec);
- si la durée du test dépasse une période de temporisation préalablement spécifiée, le résultat du test est «TIMEOUT» (expiration de la temporisation);
- si le test a pris fin à cause d'une erreur de procédure ou de la réception d'une primitive TERMINATE REQUEST (demande de fin de test), le résultat est «PREMATURE TERMINATION» (fin de test prématurée);
- dans le cas contraire, le résultat du test est «INCONCLUSIVE» (non concluant).

7.5.9 Fin de test

En mode non piloté, cette catégorie de test utilise l'arrêt de test implicite.

En mode piloté, on utilise aussi bien l'arrêt implicite qu'explicite. Pour l'arrêt explicite, on utilise le service de fin de test (TEST-TERMINATION) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

7.6 Test de limite de ressource

7.6.1 Objet de la catégorie de test

Un système peut comprendre de nombreuses parties ou ressources. L'objet de ce test est de s'assurer séparément du bon comportement des diverses ressources internes au système. Le comportement de la ressource est testé par la commande et l'observation des interactions entre la ressource et son environnement. Des points de commande et d'observation (PCO) seront placés aux limites de la ressource et les informations échangées en ces points entre les ressources à tester sont définies en termes de signaux. On teste le comportement de la ressource en insérant des signaux d'essai aux points PCO. On vérifie alors si les signaux engendrés par la ressource sont conformes à la spécification de comportement de la ressource.

Cette catégorie de test doit être prise en considération pour les ressources qui interagissent fortement avec leur environnement. On peut vérifier le comportement de la ressource lorsque ces interactions sont prescrites en termes de protocoles, de services et d'interfaces. L'Annexe K donne un exemple d'utilisation de la catégorie de test de limite de ressource.

7.6.2 Prescriptions relatives à l'objet MORT et à l'objet associé

7.6.2.1 Configuration du test de limite de ressource

L'objet MORT représente les ressources qui sont testées par le test de limite de ressource. Un objet MORT est une ressource identifiable. On peut appliquer le test à une ou plusieurs ressources adjacentes. Une ressource peut toutefois contenir, d'une manière récursive, plusieurs ressources. Lorsque plusieurs ressources sont testées au cours d'un même test, il s'agit d'un test «multiresource». Dans ce cas, chaque ressource testée est représentée par un objet MORT.

Etant donné que les points PCO sont situés aux limites des ressources, chacun d'eux peut avoir des connexions avec plusieurs ressources. Un point PCO est représenté par un objet AO. Aux points PCO, on peut observer et insérer des signaux. En outre, le flux d'informations vers les ressources adjacentes peut être déconnecté. Dans ce cas, les signaux ne sont pas retransmis aux ressources adjacentes qui ne participent pas au test. Ainsi, on évite que les données de test soient traitées comme des données d'utilisateur.

La Figure 15 montre des configurations possibles d'objets MORT et AO pour tester la limite de ressource.

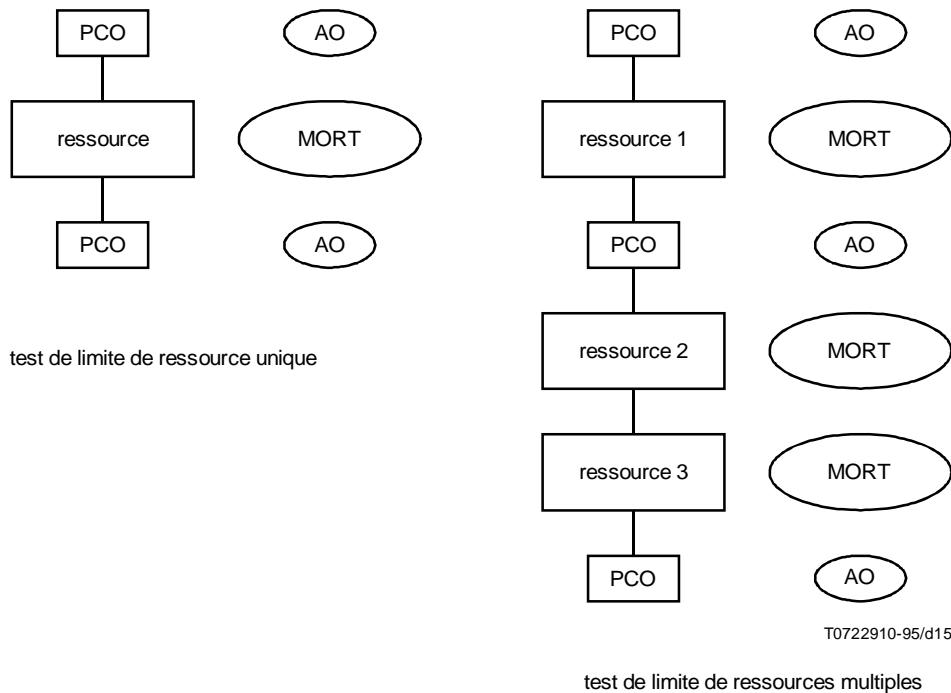


Figure 15 – Configurations possibles d'objets MORT et AO pour le test de limite de ressource

7.6.2.2 Déroulement du test

On déclenche un test de limite de ressource à l'aide d'une demande de test qui spécifie les objets MORT et AO et on y met fin à l'aide d'une demande de fin de test. Entre-temps, plusieurs événements de test peuvent être exécutés. Chaque événement de test contient un signal qui doit être inséré ou reçu à un point PCO.

La première série d'événements de test à exécuter peut être spécifiée dans la demande de test (lors de la création de l'objet «test»). On peut ensuite spécifier plusieurs fois les événements de test à exécuter par configuration de l'objet «test» en utilisant le service PT-SET (voir 9.3 de la Rec. UIT-T X.745 | ISO/CEI 10164-12, Fonction de gestion de test.

Dans le cas d'un comportement non déterministe, la séquence d'événements de test contiendra généralement un seul signal à insérer et la réponse prévue n'est alors pas spécifiée. Tout signal reçu dans ce cas doit être immédiatement retransmis au système de gestion.

Plusieurs signaux peuvent être inclus dans la séquence d'événements lorsqu'on s'attend à un comportement entièrement déterminé de la ressource à tester. Dans ce cas, les réponses prévues en termes de signaux sont également spécifiées. La séquence d'événements comprend alors aussi bien des signaux d'émission que des signaux de réception. Cette option permet une exécution plus rapide des tests et est particulièrement appropriée lors de l'exécution de tests à durée critique.

Les signaux d'émission sont les signaux qu'il faut insérer dans la ressource à tester par l'intermédiaire d'un point PCO. Les signaux de réception sont engendrés par la ressource à tester et sont visibles à un point PCO spécifié. A un point PCO, les signaux peuvent être reçus de l'une quelconque des ressources connectées ou être insérés dans cette ressource. Etant donné que les signaux reçus peuvent dévier par rapport aux résultats escomptés, le déroulement du test (interactions entre le système de gestion et le système géré) n'est pas préalablement défini.

Le réalisateur du test doit considérer la liste qui contient la séquence d'événements comme étant active dès qu'il commence à traiter les événements. La séquence d'événements sera traitée d'une manière séquentielle tant que cette liste est active. La liste restera active jusqu'à ce que tous les événements aient été traités avec succès ou qu'une défaillance se produise.

La façon de traiter les événements diffère pour les événements d'émission et de réception. Dans le cas d'un événement d'émission, le réalisateur du test doit d'abord attendre l'expiration d'un temporisateur optionnel associé à l'événement. Il insère alors le signal à un point PCO donné. Dans le cas d'un événement de réception, le réalisateur du test doit attendre la réception du signal indiqué avant de passer à l'événement suivant. La réception d'un autre signal puis du signal suivant dans la liste, ou l'expiration d'une temporisation de délai d'attente provoque une défaillance; dans ce cas, il est mis fin à la procédure de traitement de la liste.

7.6.3 Environnement de test

Le test d'intégrité des données peut être intrusif ou non par rapport aux objets MORT et AO.

Si le test est intrusif, il sera abandonné dans les conditions suivantes:

- si les objets MORT ou AO n'ont pas été préalablement placés dans un état approprié (réservation pour test par exemple);
- si les objets MORT ou AO ne peuvent être placés dans un état approprié lors du lancement du test.

7.6.4 Tests pilotés/non pilotés

Cette catégorie de test est modélisée sous la seule forme de test piloté, car de nombreuses interactions entre le système de gestion et le système géré sont nécessaires lorsqu'une ressource est testée.

7.6.5 Prescriptions d'objet TO

La présente Recommandation | Norme internationale définit un objet «test de limite de ressource». Cet objet «test» peut être spécialisé pour définir des sous-classes d'objets «tests» plus spécifiques.

7.6.6 Lancement spécifique de la catégorie de test

Tous les paramètres qui sont obligatoires dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 le sont également pour cette catégorie de test. Les paramètres suivants définis comme étant optionnels dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 sont obligatoires pour cette catégorie de test:

- identité des objets AO;
- pour chaque objet AO: son état, par exemple indiquer, pour chaque flux d'informations, s'il sera observé et s'il sera déconnecté (à l'aide du paramètre `associatedObjetInform` de la définition ASN.1 des objets AO).

Les informations complémentaires suivantes peuvent être fournies dans la demande de test (elles peuvent être également fournies dans les opérations PT-SET exécutées sur un objet TO):

- indicateur de compte rendu de résultat (pour indiquer si un compte rendu de résultat est nécessaire lorsqu'une séquence d'événements de test a été exécutée avec succès);
- séquence d'événements de test comprenant chacun les éléments suivants:
 - identificateur d'événement;
 - type de signal;
 - paramètres du signal et valeurs associées;
 - sens du signal (insertion ou réception);
 - identité de l'objet MORT dans lequel le signal est inséré ou en provenance duquel il est reçu;
 - identité de l'objet AO où le signal est inséré ou reçu;
 - valeur du temporisateur de délai d'attente (dans le cas d'un signal reçu, ce temporisateur indique le délai d'attente de ce signal et dans le cas d'un signal inséré, ce temporisateur spécifie le délai d'attente entre la fin de l'événement précédent et l'insertion du signal).

7.6.7 Événements déclenchant la notification du résultat et la fin de test

Quatre événements déclenchent la notification de résultats intermédiaires:

- cas 1 – Fin du dernier signal de la séquence d'événements de test (si elle est indiquée par l'indicateur de compte rendu de résultat);
- cas 2 – Réception d'un signal au point PCO lorsque aucune liste de séquence d'événements n'est active et qu'aucun signal de réception ne doit donc être mis en correspondance avec une telle liste;
- cas 3 – Réception d'un signal à un point PCO alors qu'un autre signal de réception doit être mis en correspondance avec la liste de séquence d'événements active;
- cas 4 – Expiration du temporisateur de délai d'attente dans le cas d'un signal de réception.

Un événement déclenche la fin du test de limite de ressource et la notification implicite du résultat final du test:

- réception d'une demande de fin de test envoyée par le conducteur du test.

7.6.8 Compte rendu du résultat du test

La catégorie de test de limite de ressource prend en charge la notification de résultat sollicitée et implicite. Le compte rendu sollicité est assuré par une demande de lecture. Le compte rendu implicite est assuré au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

Le test utilise la notification implicite car, pour le test de limite de ressource, il est important que les réponses du système géré soient transmises dès que possible au système de gestion. Il y a deux raisons à cela: premièrement, des tests à durée critique peuvent être effectués et, deuxièmement, en cas de comportement imprévu, le système de gestion ignore à quel moment des résultats de test sont disponibles.

Les informations renvoyées dans ce compte rendu dépendent de l'événement qui déclenche la notification du résultat et ne se rapportent qu'à l'événement qui déclenche l'envoi du compte rendu. Les informations suivantes relatives aux cas mentionnés au 7.6.7 doivent être fournies:

dans le cas 1:

- résultat intermédiaire du test de limite de ressource (succès).

dans le cas 2:

- résultat intermédiaire du test de limite de ressource (imprévu);
- type du signal reçu;
- valeurs du type de signal reçu;
- identité de l'objet MORT en provenance duquel le signal a été reçu;
- identité de l'objet AO où le signal a été reçu.

dans le cas 3:

- résultat intermédiaire du test de limite de ressource (wrongSignalReceived – réception d'un signal erroné);
- identificateur d'événement de test ayant échoué;
- type du signal reçu;
- valeurs du type de signal reçu;
- identité de l'objet associé où le signal a été reçu.

dans le cas 4:

- résultat intermédiaire du test de limite de ressource (waitDurationTimerExpired – expiration du temporisateur de délai d'attente);
- identificateur d'événement de test ayant échoué.

La valeur du résultat de test à renvoyer dans le résultat final est «INCONCLUSIVE» (non concluant).

7.6.9 Fin de test

Ce test utilise la fin explicite car le système de gestion commande les tests. Tous les renseignements concernant les tests sont contenus dans le système de gestion; par exemple, le déroulement du test est commandé par le système de gestion. Il faut donc que le système de gestion décide également à quel moment un test doit prendre fin.

7.7 Autotest de ressource

7.7.1 Objet de la catégorie de test

Les autotests de ressource servent à étudier l'aptitude d'une ressource (fonction ou entité physique) à assumer la fonction qui lui est assignée à un instant donné. La nature de la ressource ou des ressources à tester et les opérations à exécuter sont propres au système dont elles dépendent.

7.7.2 Prescriptions relatives à l'objet MORT et à l'objet associé

Chaque autotest de ressource implique la participation d'un seul objet MORT qui est la ressource à tester. Le test entraînera l'activation de certaines des fonctions à l'intérieur de la ressource, et donc la modification éventuelle de la valeur de certains attributs de l'objet MORT. Le conducteur du test pourra surveiller ces attributs de l'objet MORT. Cette catégorie de test ne fait pas intervenir d'objet associé. La Figure 16 représente les objets gérés utilisés par cette catégorie de test.

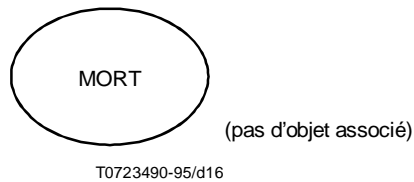


Figure 16 – Exemple d'environnement pour un autotest de ressource

7.7.3 Environnement de test

Selon la nature du test et de l'objet MORT, l'autotest de ressource peut être intrusif ou non par rapport à l'objet MORT.

Si l'autotest de ressource est intrusif, il sera abandonné dans les conditions suivantes:

- si l'objet MORT n'a pas été préalablement placé dans un état approprié (réservation pour test par exemple);
- si l'objet MORT ne peut être placé dans un état approprié lors du lancement du test.

7.7.4 Tests pilotés/non pilotés

Cette catégorie de test peut être appelée à l'aide du service de demande de test non piloté ou du service de demande de test piloté.

7.7.5 Prescriptions d'objet TO

La présente Recommandation | Norme internationale définit un objet «autotest de ressource». Cet objet «test» peut être spécialisé pour définir des classes d'objets «tests» plus spécifiques.

7.7.6 Lancement spécifique de la catégorie de test

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres optionnels suivants peuvent également être nécessaires:

- identité de l'objet MORT s'il n'est pas implicitement connu (par exemple si l'objet contenant la fonctionnalité de récepteur TARR est l'objet MORT lui-même);
- type de diagnostic à effectuer. Lorsque le type de diagnostic n'est pas mentionné dans la demande de test, c'est qu'il est particulier et connu de l'objet MORT;
- phases de déroulement du test;
- nombre d'itérations de chaque phase;
- temporisation d'exécution.

7.7.7 Événements déclenchant la notification du résultat et la fin du test

Le conducteur du test peut extraire, sous forme explicite, les résultats de l'autotest de ressource, lorsqu'ils sont disponibles, grâce à l'attribut correspondant; sinon, ces résultats peuvent être implicitement générés. Les événements qui déclenchent la notification implicite du résultat sont les suivants:

- tout instant au cours de la mise en œuvre d'une fonction, généralement à la fin des phases appropriées, si elles sont définies;
- achèvement de mise en œuvre de la fonction.

Au cours de l'exécution de l'autotest de ressource, le conducteur du test peut mettre fin explicitement ou implicitement à ce test. Les événements qui déclenchent la fin de test implicite sont les suivants:

- achèvement du test;
- échec de la mise en œuvre normale du test;
- expiration de l'intervalle de temps pendant lequel le test doit être achevé.

7.7.8 Compte rendu du résultat du test

Lorsque le test obéit à un modèle non piloté, le compte rendu implicite sera utilisé et viendra en réponse à une ou plusieurs actions. Lorsque le test obéit à un modèle piloté, le compte rendu sollicité sera pris en charge et sera déclenché par une demande de lecture; le compte rendu conditionnellement implicite peut également être utilisé au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres suivants sont nécessaires:

- identité de l'objet MORT si elle n'est pas incluse dans la demande de test;

Les paramètres optionnels suivants peuvent être également renvoyés:

- informations spécifiques des exercices définis;
- le cas échéant, autres informations définissant aussi exactement que possible toute action complémentaire qui devrait être éventuellement entreprise;
- le résultat du test choisi selon le cas en fonction de la Rec. UIT-T X.745 | ISO/CEI 10164-12 doit être renvoyé dans le compte rendu de résultat final.

La valeur du résultat du test à renvoyer dans le compte rendu de test final est déterminée comme suit:

- si toutes les procédures de test ont été exécutées avec succès, le résultat du test est «PASS» (succès);
- si une procédure de test quelconque détecte une erreur, le résultat est «FAIL» (échec);
- si la durée du test dépasse une valeur de temporisation préalablement spécifiée, le résultat du test est «TIMEOUT» (expiration de la temporisation);
- si le test a pris fin à cause d'une erreur de procédure ou de la réception d'une primitive TERMINATE REQUEST (demande de fin de test), le résultat est «PREMATURE TERMINATION» (fin de test prématurée);
- dans les autres cas, le résultat du test est «INCONCLUSIVE» (non concluant).

7.7.9 Fin de test

En mode non piloté, cette catégorie de test utilise l'arrêt de test implicite.

En mode piloté, on utilise aussi bien l'arrêt implicite qu'explicite. Pour l'arrêt explicite, on utilise le service de fin de test (TEST-TERMINATION) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

7.8 Test d'infrastructure d'essai

7.8.1 Objet de la catégorie de test

Les tests d'infrastructure d'essai permettent d'étudier l'aptitude d'un système ouvert géré à déclencher des tests, à renvoyer des comptes rendus de résultat et (dans le cas de tests pilotés) à répondre à des actions de contrôle et de commande. Par exemple, on peut effectuer un test d'infrastructure d'essai avant de déclencher des tests de ressources spécifiques mettant en jeu le système ouvert géré.

Le test d'infrastructure d'essai est un test «nul» dont le but est uniquement d'étudier l'aptitude d'un système ouvert géré à se comporter comme indiqué dans la Rec. UIT-T X.745 | ISO/CEI 10164-12, c'est-à-dire recevoir une demande de test entrante et engendrer des réponses appropriées. Aucun exercice de test n'est effectué sur des ressources internes ou externes au système ouvert géré.

Les passages successifs aux états de test définis par la Rec. UIT-T X.745 | ISO/CEI 10164-12 auront lieu à intervalles spécifiés. Si un test quelconque offert par le système ouvert géré permet de déterminer l'aptitude:

- à répondre à des demandes de suspension/reprise;
- à répondre à des demandes de fin de test;
- à effectuer des essais programmés;
- à abandonner les tests après des périodes de temporisation spécifiées.

Le test d'infrastructure d'essai devrait permettre de vérifier ces aspects de mise en œuvre de la Rec. UIT-T X.745 | ISO/CEI 10164-12 sur le système ouvert géré.

NOTE – Lorsqu'on utilise cette catégorie de test, il est nécessaire de spécifier la signification exacte de la progression à travers les différents états du test: les «réponses appropriées», les états pris en charge, le temps passé dans chaque état, etc.

7.8.2 Prescriptions relatives à l'objet MORT et à l'objet associé

L'objet MORT est implicitement l'objet géré comportant la fonctionnalité TARR. Aucun objet associé AO n'intervient dans ce test. La Figure 17 représente les objets gérés intervenant dans cette catégorie de test.

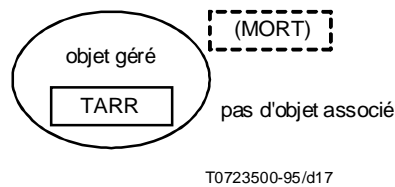


Figure 17 – Exemple d'environnement pour un test d'infrastructure d'essai

7.8.3 Environnement de test

Le test d'infrastructure d'essai est de type non intrusif.

7.8.4 Tests pilotés/non pilotés

Ce test peut être modélisé sous la forme d'un test piloté ou non piloté.

7.8.5 Prescriptions d'objet «test» TO

La présente Recommandation | Norme internationale définit un objet «test d'infrastructure d'essai». Cet objet «test» TO ne peut être spécialisé.

7.8.6 Lancement spécifique de la catégorie de test

Lorsqu'un test non piloté est demandé, le paramètre de période de temporisation défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 doit être spécifié.

Lorsqu'un test piloté est demandé, seuls les paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12 doivent être spécifiés.

L'intervalle de temps nécessaire entre les transitions d'états de test peut de plus être spécifié (valeur par défaut spécifique de la mise en œuvre si cet intervalle n'est pas indiqué).

7.8.7 Événements déclenchant la notification du résultat et la fin de test

Les événements qui déclenchent une notification implicite de résultat sont les suivants:

- transition entre les états de test (si la notification est prise en charge dans ce cas);
- achèvement de l'exercice.

Les événements qui déclenchent l'arrêt de test implicite sont les suivants:

- achèvement de l'exercice;
- expiration de la temporisation;
- non-exécution du test avec succès à la suite d'une erreur.

7.8.8 Compte rendu du résultat du test

Lorsque le test obéit à un modèle non piloté, le compte rendu implicite sera utilisé et viendra en réponse à une ou plusieurs actions. Lorsque le test obéit à un modèle piloté, le compte rendu sollicité sera pris en charge et sera déclenché par une demande de lecture; le compte rendu conditionnellement implicite peut également être utilisé au moyen du service de résultat de test (TEST-RESULT) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 (désigné également par le terme de compte rendu non sollicité).

En plus des paramètres requis par la Rec. UIT-T X.745 | ISO/CEI 10164-12, les paramètres suivants sont également nécessaires:

- s'il s'agit du compte rendu final de test, le résultat du test choisi comme indiqué ci-dessous.

Le paramètre optionnel suivant peut également être renvoyé:

- autres informations spécifiques de la mise en œuvre du test;

Le résultat du test est déterminé comme suit:

- si tous les exercices ont été exécutés avec succès, le résultat du test est «PASS» (succès);
- si un exercice quelconque a détecté une erreur, le résultat est «FAIL» (échec);
- si la durée du test a dépassé une valeur de temporisation spécifiée, le résultat du test est «Timed-out» (expiration de la temporisation);
- si le test a pris fin à cause d'une erreur de procédure ou de la réception d'une primitive TERMINATION REQUEST (demande de fin de test), le résultat est «PREMATURE TERMINATION» (fin de test prématurée);
- dans le cas contraire, le résultat sera «INCONCLUSIVE» (non concluant).

7.8.9 Fin de test

En mode non piloté, cette catégorie de test utilise l'arrêt de test implicite.

En mode piloté, on utilise aussi bien l'arrêt implicite qu'explicite. Pour l'arrêt explicite, on utilise le service de fin de test (TEST-TERMINATION) défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12.

8 Définitions génériques

Cet article spécifie un ensemble d'attributs et/ou de paramètres génériques appropriés à une intégration dans la définition de classes d'objets TO, dans des notifications de résultat de test ou pouvant servir de paramètres d'appel de test. Les modèles de ces définitions sont reproduits dans l'Annexe A.

8.1 Types d'attributs génériques

8.1.1 Résultats de test de connexion

Cet attribut contient les résultats du test de connexion.

8.1.2 Résultats du test de connectivité

Cet attribut contient les résultats du test de connectivité.

8.1.3 Type de test de connectivité

Cet attribut indique si le test de connectivité se rapporte à un protocole en mode connexion ou à un protocole en mode sans connexion.

8.1.4 Type de test d'intégrité des données

Cet attribut indique si le test d'intégrité des données se rapporte à un protocole en mode connexion ou à un protocole en mode sans connexion.

8.1.5 Résultats de test d'intégrité des données

Cet attribut contient les résultats du test d'intégrité des données. Si le test a réussi, les résultats de test indiquent le temps d'établissement et l'unité de données d'origine.

8.1.6 Unités de données

Cet attribut sert à spécifier le type et le volume d'unités de données de test à envoyer au cours du test. Il n'est utilisé que si le test prend en charge la spécification de module de gestion du type et/ou du volume d'unités de données.

8.1.7 Résultats du test de connexion terminale

Cet attribut contient les résultats du test de connexion terminale.

8.1.8 Type de rapport de taux d'erreur

Cet attribut indique le mode de mesure du taux d'erreur, qui peut se présenter sous la forme du nombre de bits erronés ou d'un pourcentage de secondes erronées.

8.1.9 Temps d'établissement

Cet attribut indique le délai nécessaire pour établir la connectivité entre deux entités.

8.1.10 Résultats du test de bouclage

Cet attribut indique les résultats d'un test de bouclage, y compris la durée du test si elle est spécifiée, les unités de données utilisées au cours du test, le taux d'erreur et, si des erreurs ont été constatées, leur cause.

8.1.11 Seuil de bouclage

Cet attribut spécifie le seuil d'erreur. En cas de dépassement de ce seuil, le test de bouclage associé doit prendre fin et le résultat du test indiqué est «FAIL» (échec).

8.1.12 Type de bouclage

Cet attribut ou paramètre sert à indiquer le type de bouclage (bouclage physique, écho, analogique ou numérique, bouclage physique non transparent, bouclage physique transparent, bouclage physique de la charge utile, renvoi en écho des données, etc.).

8.1.13 Réception d'unité PDU

Cet attribut contient l'information d'unité PDU qui sert à effectuer une comparaison avec l'unité PDU issue d'un test d'intégrité de protocole afin de déterminer quel élément de protocole il y a lieu d'envoyer lors de la transmission suivante et à quel moment celle-ci doit avoir lieu.

8.1.14 Séquence d'unités PDI

Cet attribut contient la séquence d'unités PDU pour un test d'intégrité de protocole.

8.1.15 Résultats du test d'intégrité de protocole

Cet attribut indique la cause de l'échec d'un test d'intégrité de protocole. Un tel échec peut être dû à une réponse par unité PDU erronée ou à l'absence d'accusé de réception d'unité PDU.

8.1.16 Résultats du test de limite de ressource

Cet attribut contient des renseignements détaillés sur l'échec d'un test et/ou sur le signal reçu à un point PCO si ce signal n'était pas spécifié comme faisant partie d'une séquence d'événements. Deux types d'échec de test peuvent être notifiés:

- un signal erroné, c'est-à-dire un signal autre que le signal spécifié dans la séquence d'événements, a été reçu;
- le temporisateur de délai d'attente a expiré: un signal prévu a été spécifié mais aucun signal n'a été reçu.

Dans les deux cas, l'identificateur de l'événement de test ayant échoué est notifié.

8.1.17 Résultats de l'autotest de ressource

Cet attribut contient les résultats d'un autotest de ressource qui indiquent la fonction testée et, le cas échéant, tout résultat complémentaire relatif à cette fonction.

8.1.18 Périodicité de compte rendu

Cet attribut spécifie l'intervalle entre les comptes rendus de résultat. Un nouveau compte rendu est produit à chaque intervalle spécifié après le lancement du test.

8.1.19 Indicateur de compte rendu de résultat

Cet attribut indique si un compte rendu de résultat est nécessaire lorsqu'une séquence d'événements de test a été exécutée.

8.1.20 Séquence d'événements

Cet attribut spécifie une séquence de signaux qui doivent être insérés ou reçus au niveau de l'objet MORT ou AO spécifié. Cette séquence est appelée «séquence d'événements de test». Elle peut contenir un seul signal qui doit être inséré ou reçu mais une séquence d'événements comprenant plusieurs signaux d'émission et de réception peut être également spécifiée. Le réalisateur du test doit traiter les événements d'une manière séquentielle. En cas d'événement de réception, le réalisateur du test doit attendre, jusqu'à l'expiration du temporisateur de délai d'attente, la réception du signal indiqué avant de passer à l'événement suivant. En cas d'expiration, une défaillance a pu se produire dans la ressource. En cas de signal d'émission, le temporisateur de délai d'attente spécifie le délai d'attente entre la fin de l'événement précédent et l'insertion du signal.

8.1.21 Conditions de test

Cet attribut spécifie les conditions dans lesquelles il convient d'affecter les ressources au test. Il indique en particulier s'il y a lieu de lancer le test lorsque l'objet MORT est à l'état d'occupation et si l'utilisateur de cet objet peut provoquer l'abandon de ce test.

8.1.22 Signal de test

Cet attribut identifie un type de signal particulier.

8.1.23 Séquence de signal de test

Cet attribut contient les données à utiliser pour un test de bouclage.

8.1.24 Seuil de taux d'erreur du test

Cet attribut spécifie le seuil du taux d'erreur. Lorsque ce seuil est dépassé, le test associé doit prendre fin et le résultat du test doit être «FAIL» (échec).

8.1.25 Intervalle d'attente

Cet attribut indique, le cas échéant, l'intervalle d'attente avant la transmission de l'élément de protocole suivant. Cet intervalle commence au moment où le dernier élément de protocole a été émis.

8.2 Objets gérés**8.2.1 Objet «test de connexion»**

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Cette classe d'objets peut être instanciée pour représenter les caractéristiques d'un test de connexion.

8.2.2 Objet «test de connectivité»

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Cette classe d'objets peut être instanciée pour représenter les caractéristiques d'un test de connectivité. S'il est géré, le bloc de notification de résultat de test peut admettre le compte rendu non sollicité. Si la terminaison implicite est gérée par l'objet géré qui possède une fonction de récepteur TARR, un objet géré de test de connectivité peut être supprimé implicitement lors de la terminaison de ce test.

8.2.3 Objet «test d'intégrité des données»

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Cette classe d'objets peut être instanciée pour prendre en charge un test d'intégrité des données. Cette classe d'objets gérés représente les caractéristiques d'un test d'intégrité des données. Elle est créée en conséquence de la réception d'un service de demande de test piloté tel que défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Si l'objet géré qui possède la fonction de récepteur TARR admet la terminaison implicite, un objet géré de test d'intégrité des données peut être supprimé implicitement lors de la terminaison de ce test.

8.2.4 Objet «autotest de ressource»

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Cette classe d'objets peut être instanciée pour prendre en charge un autotest de ressource. Cette classe d'objets gérés représente les caractéristiques d'un autotest de ressource. Cette classe d'objets peut être instanciée à la suite de la réception d'un service de demande de test piloté tel que défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Si l'objet géré qui possède la fonction de récepteur TARR peut admettre la terminaison implicite, un objet géré d'autotest de ressource peut être supprimé implicitement lors de la terminaison de ce test.

8.2.5 Objet «test de bouclage»

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Cette classe d'objets peut être instanciée pour prendre en charge un test de bouclage. Cette classe d'objets gérés représente les caractéristiques d'un test de bouclage. Cette classe d'objets peut être instanciée en conséquence de la réception d'un service de demande de test piloté tel que défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Si l'objet géré qui possède la fonction de récepteur TARR peut admettre la terminaison implicite, un objet géré de test de bouclage peut être supprimé implicitement lors de la terminaison de ce test.

8.2.6 Objet «test d'intégrité de protocole»

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Cette classe d'objets peut être instanciée pour prendre en charge un test d'intégrité de protocole. Cette classe d'objets gérés représente les caractéristiques d'un test d'intégrité de protocole. Cette classe d'objets peut être instanciée en conséquence de la réception d'un service de demande de test piloté tel que défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Si l'objet géré qui possède la fonction de récepteur TARR peut admettre la terminaison implicite, un objet géré de test d'intégrité de protocole peut être supprimé implicitement lors de la terminaison de ce test.

8.2.7 Objet «test de limite de ressource»

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Elle est instanciée pour prendre en charge un test de limite de ressource. Cette classe d'objets gérés représente les caractéristiques de gestion d'un test de limite de ressource. Elle peut être instanciée à la suite de la réception d'un service de demande de test piloté défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Etant donné que le paquetage de notification de résultat de test est présent, les résultats du test peuvent être notifiés implicitement. Un objet «test de limite de ressource» ne peut être supprimé qu'explicitement.

8.2.8 Objet «test d'infrastructure d'essai»

Cette classe d'objets gérés dérive de la classe d'objets gérés d'objet «test» définie dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Elle est instanciée pour prendre en charge un test d'infrastructure d'essai. Cette classe d'objets gérés représente les caractéristiques de gestion d'un test d'infrastructure d'essai. Elle peut être instanciée à la suite de la réception d'un service de demande de test piloté défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Si l'objet géré qui possède la fonction de récepteur TARR prend en charge la terminaison implicite, un objet géré de test d'infrastructure d'essai peut être supprimé implicitement lors de la terminaison de ce test.

8.3 Définition générique importée

La présente Recommandation | Norme internationale utilise les définitions suivantes, issues de la Rec. UIT-T X.745 | ISO/CEI 10164-12:

- paquetage d'objets MORT;
- paquetage d'objets associés AO;
- service de notification de résultat de test;
- paquetage de temporisation;
- paquetage de conflit de planification.

NOTE – Toute classe d'objets instanciable comportant l'information optionnelle d'objet associé doit utiliser un gabarit de paramètres applicables à l'attribut «associatedObjects» (objets associés) à l'action «testRequestControlledAction» (Action pilotée de demande de test) et/ou «testRequestUncontrolledAction» (Action non pilotée de demande de test). Il convient par ailleurs d'utiliser le mot clé du contexte «associatedObjects.AssociatedObjectInfo» (Objets associés.Info d'objet associé) dans le gabarit de paramètres qui sert à enregistrer la syntaxe et l'identificateur d'objet pour «ANY DEFINED BY».

8.4 Compatibilité

Les définitions de classe d'objets gérés prennent en charge les catégories de test définies dans la présente Recommandation | Norme internationale en incorporant une référence à la définition appropriée des classes d'objets gérés, des attributs, des actions et des notifications, figurant dans l'Annexe A de la présente Recommandation | Norme internationale, dans l'Annexe A de la Rec. UIT-T X.745 | ISO/CEI 10164-12 et dans la Rec. X.721 du CCITT | ISO/CEI 10165-2. Le mécanisme de référence est défini dans la Rec. X.722 du CCITT | ISO/CEI 10165-4.

9 Services

La présente Recommandation | Norme internationale fait appel aux services définis dans d'autres fonctions de gestion-système. Elle ne définit pas de services additionnels.

10 Unités fonctionnelles

La présente Recommandation | Norme internationale fait appel aux unités fonctionnelles définies dans la Rec. UIT-T X.745 | ISO/CEI 10164-12 et ne définit pas d'unités fonctionnelles supplémentaires.

11 Protocole

La spécification du protocole applicable à la présente Recommandation | Norme internationale est la même que pour la Rec. UIT-T X.745 | ISO/CEI 10164-12.

12 Relations avec d'autres fonctions

Les catégories de test définies dans la présente Recommandation | Norme internationale sont fondées sur le modèle défini dans la Rec. UIT-T X.745 | ISO/CEI 10164-12. Si le compte rendu des résultats d'essai est de type explicite, cette Recommandation | Norme internationale fait appel au service de transfert d'obtentions (PT-GET) défini dans la Rec. X.730 du CCITT | ISO/CEI 10164-1. Lors du changement des valeurs d'attribut dans les objets «tests», la présente Recommandation | Norme internationale utilise le service de transfert de modifications (PT-SET) défini dans la Rec. X.730 du CCITT | ISO/CEI 10164-1.

13 Conformité

Les réalisations déclarées conformes à la présente Recommandation | Norme internationale respecteront les prescriptions de conformité définies ci-après.

13.1 Conformité statique

La réalisation sera conforme aux prescriptions de la présente Recommandation | Norme internationale dans le rôle de gestionnaire, dans le rôle d'agent ou dans les deux. Une déclaration de conformité à au moins l'un de ces rôles sera effectuée dans le Tableau B.1.

Si une déclaration de conformité est effectuée pour le rôle de gestionnaire, la réalisation prendra en charge au moins une des opérations de gestion d'attribut ou une des notifications d'objets gérés définies dans la présente Recommandation | Norme internationale. Les prescriptions de conformité dans le rôle de gestionnaire pour ces opérations de gestion sont identifiées dans le Tableau B.2 et dans d'autres tableaux cités en référence dans l'Annexe B.

Si une déclaration de conformité est effectuée pour le rôle d'agent, la réalisation prendra en charge une ou plusieurs instances des objets gérés décrits dans le Tableau B.3. Une déclaration de conformité dans le rôle d'agent nécessite la prise en charge de toutes les opérations et notifications obligatoires spécifiées par ces définitions de gestion.

La réalisation prendra en charge la syntaxe de transfert dérivée des règles de codage spécifiées dans la Rec. UIT-T X.209 | ISO/CEI 8825 appelées {joint-iso-ccitt asn1 (1) basicEncoding(1)} pour les types de données abstraites cités en référence par les définitions dont la prise en charge est revendiquée.

13.2 Conformité dynamique

Les applications déclarées conformes à la présente Recommandation | Norme internationale prendront en charge les éléments de procédure et la sémantique des définitions dont la prise en charge est revendiquée.

13.3 Prescriptions relatives à la déclaration de conformité d'instance de gestion

Tout formulaire MCS, MOCS et MRCS conforme à la présente Recommandation | Norme internationale sera techniquement identique aux formulaires spécifiés dans les Annexes B, C et D, en conservera la numérotation des tableaux et des rubriques, et n'en différera que par la pagination et les en-têtes de page.

Dans le cadre des prescriptions de conformité, le fournisseur d'une réalisation déclarée conforme à la présente Recommandation | Norme internationale remplira un exemplaire du récapitulatif de conformité de gestion (MCS) fourni dans l'Annexe B, ainsi que tout autre formulaire d'ICS mentionné s'il y a lieu par ce formulaire de MCS. Un formulaire de déclaration ICS conforme à la présente Recommandation | Norme internationale:

- décrira une application conforme à la présente Recommandation | Norme internationale;
- aura été rempli conformément aux instructions données dans la Rec. UIT-T X.724 | ISO/CEI 10165-6;
- comportera les informations nécessaires pour identifier sans ambiguïté le fournisseur et l'application.

Annexe A

Définition de l'information de gestion

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

A.1 Test category related management information

For each test category, this subclause summarizes the management information specified for the category.

In the following subclauses for each test category, the parameters associated with the test category information parameter(s) have assigned object identifiers (see A.5) which are used to identify that particular test category in the test request action information, test request action reply (for controlled tests) and test request notification reply (for controlled tests).

A.1.1 Connection test category

- a) Test request service type: testRequestControlledAction and/or testRequestUncontrolledAction.
- b) Specific errors:
 - Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable, independentTestInvocationError, relatedTOError.
 - Test request uncontrolled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testSuspendResumeError.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.
- c) Test category information parameter:
 - Test request controlled action information parameter: connectionTestInfoParam.
 - Test request uncontrolled action information parameter: connectionTestInfoParam.
 - Test request notification reply parameter: connectionControlledResultsParam.
 - Test request uncontrolled action reply parameter: connectionUncontrolledResultsParam.
- d) Test control: Test suspend/resume and Test termination are applicable if controlled test is in use.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: connectionTestObject.

A.1.2 Connectivity test category

- a) Test request service type: testRequestControlledAction and/or testRequestUncontrolledAction.
- b) Specific errors:
 - Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable, independentTestInvocationError, relatedTOError.
 - Test request uncontrolled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.

- c) Test category information parameter:
 - Test request controlled action information parameter: connectivityTestInfoParam.
 - Test request uncontrolled action information parameter: connectivityTestInfoParam.
 - Test request notification reply parameter: connectivityControlledResultsParam, invalidTestOperation, testSuspendResumeError.
 - Test request uncontrolled action reply parameter: connectivityUncontrolledResultsParam.
- d) Test control: Test suspend/resume and Test termination are applicable if controlled test is in use.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: connectivityTestObject.

A.1.3 Data integrity test category

- a) Test request service type: testRequestControlledAction and/or testRequestUncontrolledAction.
- b) Specific errors:
 - Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable, independentTestInvocationError, relatedTOError.
 - Test request uncontrolled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testSuspendResumeError.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.
- c) Test category information parameter:
 - Test request controlled action information parameter: dataIntegrityTestInfoParam.
 - Test request uncontrolled action information parameter: dataIntegrityTestInfoParam.
 - Test request notification reply parameter: dataIntegrityControlledResultsParam.
 - Test request uncontrolled action reply parameter: dataIntegrityUncontrolledResultsParam.
- d) Test control: Test suspend/resume and Test termination are applicable if controlled test is in use.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: dataIntegrityTestObject.

A.1.4 Loopback test category

- a) Test request service type: testRequestControlledAction and/or testRequestUncontrolledAction.
- b) Specific errors:
 - Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable, independentTestInvocationError, relatedTOError.
 - Test request uncontrolled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.
- c) Test category information parameter:
 - Test request controlled action information parameter: loopbackTestInfoParam.
 - Test request uncontrolled action information parameter: loopbackTestInfoParam.
 - Test request notification reply parameter: loopbackControlledResultsParam.
 - Test request uncontrolled action reply parameter: loopbackUncontrolledResultsParam.

- d) Test control: Test suspend/resume and Test termination are applicable if controlled test is in use.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: loopbackTestObject.

A.1.5 Protocol integrity test category

- a) Test request service type: testRequestControlledAction and/or testRequestUncontrolledAction.
- b) Specific errors:
 - Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable, independentTestInvocationError, relatedTOError.
 - Test request uncontrolled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testSuspendResumeError.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.
- c) Test category information parameter:
 - Test request controlled action information parameter: protocolIntegrityTestInfoParam.
 - Test request uncontrolled action information parameter: protocolIntegrityTestInfoParam.
 - Test request notification reply parameter: protocolIntegrityControlledResultsParam.
 - Test request uncontrolled action reply parameter: protocolIntegrityUncontrolledResultsParam.
- d) Test control: Test suspend/resume and Test termination are applicable if controlled test is in use.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: protocolIntegrityTestObject.

A.1.6 Resource Boundary test category

- a) Test request service type: testRequestControlledAction.
- b) Specific Errors:
 - Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, noSuchAssociatedObject, associatedObjectNotAvailable, independentTestInvocationError, relatedTOError.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testSuspendResumeError.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.
- c) Test category information parameter:
 - Test request controlled action information parameter: resourceBoundaryTestInfoParam.
NOTE – No ‘Test request notification reply parameter’ is defined, because no test category specific reply information is needed for the resource boundary test.
- d) Test control: Test suspend/resume and Test termination are applicable.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: resourceBoundaryTestObject.

A.1.7 Resource self test category

- a) Test request service type: testRequestControlledAction and/or testRequestUncontrolledAction.
- b) Specific errors:
 - Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, independentTestInvocationError, relatedTOError.

- Test request uncontrolled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testSuspendResumeError.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.
- c) Test category information parameter:
- Test request controlled action information parameter: resourceSelfTestInfoParam.
 - Test request uncontrolled action information parameter: resourceSelfTestInfoParam.
 - Test request notification reply parameter: resourceSelfControlledResultsParam.
 - Test request uncontrolled action reply parameter: resourceSelfUncontrolledResultsParam.
- d) Test control: Test suspend/resume and Test termination are applicable if controlled test is in use.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: resourceSelfTestObject.

A.1.8 Test infrastructure test category

- a) Test request service type: testRequestControlledAction and/or testRequestUncontrolledAction.
- b) Specific errors:
- Test request controlled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId, independentTestInvocationError, relatedTOError.
 - Test request uncontrolled action: noSuchMORT, mORTNotAvailable, mistypedTestRequestInformation, noSuchTestCategoryId.
 - Test suspend/resume action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testSuspendResumeError.
 - Test termination action: noSuchInvocationId, noSuchTestSessionId, invalidTestOperation, testTerminateError.
- c) Test category information parameter:
- No test category specific parameter is required.
- d) Test control: Test suspend/resume and Test termination are applicable if controlled test is in use.
- e) TO requirements: The controlled test must instantiate an instance of TO.
- f) TO class: «ITU-T Rec. X.745 (1993) | ISO/IEC 10164-12:1994»: testObject.

A.2 Generic Object Classes

A.2.1 Connection Test Object

connectionTestObject MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;

CHARACTERIZED BY

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTsPackage,
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectsPackage,

connectionTestPkg

PACKAGE

BEHAVIOUR connectionTestBehaviour **BEHAVIOUR**

DEFINED AS "The MORT(s) represent the communications path to be tested. The two associated objects represent the resources at the ends of the communication path that drive signals into and receive signals from the communication path. The test patterns represents the signals or data to be applied to the communication path. If the test is intrusive, the administrativeState and availabilityStatus may need to be supported and to be set to a proper state before the test may be initiated.";;

ATTRIBUTES

testPatterns **GET**,
connectionTestResults **GET**;;;

CONDITIONAL PACKAGES

timeoutPeriodPkg PRESENT IF !The timeout period is specified in the test request.!,
connectionTestResultPackage PACKAGE NOTIFICATIONS
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification connectionControlledResultsParam;;
PRESENT IF !Unsolicited reporting is supported.!,

REGISTERED AS {part14ObjectClass ??};

A.2.2 Connectivity Test Object

connectivityTestObject MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;

CHARACTERIZED BY

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTsPackage,
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectsPackage,
connectivityTestPkg

PACKAGE

BEHAVIOUR connectivityTestBehaviour BEHAVIOUR

DEFINED AS "The MORT represents the entity that initiates the establishment of the connection with the associated object. The connectivityTestResults and connectivityType must be included as part of the monitoredAttributes parameter of the testResultNotification. If the timeoutPeriod is present and expires before the test outcome can be concluded, the timeoutPeriod will be returned in the test result with the test outcome set to TIMEOUT.";;

ATTRIBUTES

connectivityType GET,
connectivityResults GET,
establishmentTime GET;;;

CONDITIONAL PACKAGES

timeoutPeriodPkg PRESENT IF !The timeout period is specified in the test request.!,
connectivityTestResultPackage PACKAGE NOTIFICATIONS
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification connectivityControlledResultsParam;;
PRESENT IF !Unsolicited reporting is supported.!,

REGISTERED AS {part14ObjectClass ??};

A.2.3 Data Integrity Test Object

dataIntegrityTestObject MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;

CHARACTERIZED BY

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTsPackage,
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectsPackage,
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":actualTestTimePackage,
dataIntegrityResultsPkg,
dataIntegrityTestPkg

PACKAGE

BEHAVIOUR dataIntegrityTestBehaviour BEHAVIOUR

DEFINED AS "While in the Testing state the MORT transmits data to an Associated object. The Associated object, upon reception of the data, reflects a copy of the data back to the MORT.";;

ATTRIBUTES

dataIntegrityType GET,
testPatterns GET-REPLACE;;;

CONDITIONAL PACKAGES

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":timeoutPeriodPkg PRESENT IF !The timeout period is specified in the test request.!,
dataIntegrityTestResultPackage PACKAGE NOTIFICATIONS
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification dataIntegrityControlledResultsParam;;
PRESENT IF !Unsolicited reporting is supported.!,
dataIntegrityThresholdPkg PRESENT IF !testThreshold is present in the initAttributeList parameter of the testRequestControlled request.!,

REGISTERED AS {part14ObjectClass ??};

A.2.4 Loopback Test Object

loopbackTestObject MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;

CHARACTERIZED BY

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTsPackage,
loopbackResultsPkg,
loopbackTestPkg
PACKAGE

BEHAVIOUR loopbackTestBehaviour BEHAVIOUR

DEFINED AS "The mORTs attribute identifies a part of the communications path that is to be tested. The associatedObjects attribute identifies the loopback point. The startTime and stopTime attributes (when present) indicate the time that the manager desires that the test phase be started and stopped, respectively. The loopbackResults, loopbackType and testPattern attributes must be included as part of the monitoredAttributes parameter of the testResultNotification. The testThreshold should be included if it is present in the loopback test object. The errorRatioReportType attribute indicates in what form the test conductor wants the error ratio be reported, whether in the form of number of error bits or in percentage of error seconds.";;

ATTRIBUTES

loopbackType GET-REPLACE,
testPatterns GET-REPLACE,
errorRatioReportType GET;;;

CONDITIONAL PACKAGES

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectsPackage PRESENT IF !The AO parameter is specified in the test request.!,

loopbackTestResultPackage PACKAGE NOTIFICATIONS

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification loopbackControlledResultsParam;;
PRESENT IF !Unsolicited reporting is supported.!,

timeoutPeriodPkg PRESENT IF !The timeout period is specified in the test request.!,

testConditionsPkg PRESENT IF !testConditions is present in the initAttributeList parameter of the testRequestControlled request.!,

dataUnitsPkg PRESENT IF !the capability is implemented and parameter is present in the initAttributeList parameter of the testRequestControlled request.!,

resultIntervalPkg PRESENT IF !the capability is implemented and the parameter is present in the initAttributeList parameter of the testRequestControlled request.!,

loopbackTimeoutPkg PRESENT IF !loopbackTimeout is present in the test request of the testRequestControlled request.!,

loopbackThresholdPkg PRESENT IF !loopbackThreshold is present in the initAttributeList parameter of the testRequestControlled request.!,

REGISTERED AS {part14ObjectClass ??};

A.2.5 Protocol Integrity Test Object

protocolIntegrityTestObject MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;

CHARACTERIZED BY

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTsPackage,
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectsPackage,
"Rec. X.721 (1992) | ISO/IEC 10165-2:1992":actualTestTimePackage,
protocolIntegrityResultsPackage,
protocolIntegrityTestPkg
PACKAGE

BEHAVIOUR protocolIntegrityTestBehaviour BEHAVIOUR

DEFINED AS "The MORT transmits Protocol Elements to an Associated object. The Associated object exhibits its normal behaviour in reaction to the Protocol Elements received. The responses received by the MORT may be compared to the expected responses.";;

ATTRIBUTES

pDUSequence GET-REPLACE;;;

CONDITIONAL PACKAGES

protocolIntegrityTestResultPackage PACKAGE NOTIFICATIONS

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification protocolIntegrityControlledResultsParam;;
PRESENT IF !Unsolicited reporting is supported.!,

timeoutPeriodPkg PRESENT IF !The timeout period is specified in the test request.!,

waitingIntervalPackage PRESENT IF !The waitingInterval is specified in the protocolIntegrityTestInfo.!,

pDUReceptionPackage PRESENT IF !The waitingInterval is not specified in the protocolIntegrityTestInfo.!,
 testConditionsPkg PRESENT IF !testConditions is present in the initAttributeList parameter of the
 testRequestControlled request.!

REGISTERED AS {part14ObjectClass ??};

A.2.6 Resource Boundary Test Object

resourceBoundaryTestObject MANAGED OBJECT CLASS
 DERIVED FROM "ITU-T Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;
 CHARACTERIZED BY

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTsPackage,
 "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectsPackage,
 resourceBoundaryTestResultPackage PACKAGE NOTIFICATIONS
 "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification
 resourceBoundaryControlledResultsParam;;

resourceBoundaryTestPkg
 PACKAGE

BEHAVIOUR resourceBoundaryTestBehaviour BEHAVIOUR

DEFINED AS "The mORT(s) attribute identifies the (part of a) resource(s) that will be tested. The Associated Objects identify the Points of Control and Observation at which the signals of the resource under test are observed and inserted. The signals that are inserted or expected as a response are indicated in the sequenceOfEvents attribute. The resultReportIndicator attribute indicates whether a result report is required when a sequence of test events was passed. The Resource boundary test results are defined by the resourceBoundaryControlledResultsParam parameter of the testResultNotification.";;

ATTRIBUTES

resultReportIndicator GET-REPLACE,
 sequenceOfEvents GET-REPLACE,
 resourceBoundaryTestResults GET;;;

CONDITIONAL PACKAGES

timeoutPeriodPkg PRESENT IF !The timeout period is specified in the test request.!,
 testConditionsPkg PRESENT IF !testConditions is present in the initAttributeList parameter of the
 testRequestControlledRequest.!

REGISTERED AS {part14ObjectClass ??};

A.2.7 Resource Self Test Object

resourceSelfTestObject MANAGED OBJECT CLASS
 DERIVED FROM "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;
 CHARACTERIZED BY

resourceSelfTestPkg PACKAGE

BEHAVIOUR resourceSelfTestBehaviour BEHAVIOUR

DEFINED AS "Resource self tests (sometimes termed self-test) are used to investigate the ability of a resource (function or physical entities) to perform its allotted function at a given time. The nature of the resource(s) under test and the exercises involved is specific to the containing system.";;

ATTRIBUTES resourceSelfTestResults;;

CONDITIONAL PACKAGES

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTsPackage PRESENT IF MORT id instance other
 than object with TARR functionality.!,

resourceSelfTestResultPackage PACKAGE NOTIFICATIONS

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification resourceSelfControlledResultsParam;;
 PRESENT IF !Unsolicited reporting is supported.!,

timeoutPeriodPkg PRESENT IF !The timeout period is specified in the test request.!,

testConditionsPkg PRESENT IF !testConditions is present in the initAttributeList parameter of the
 testRequestControlled request.!

REGISTERED AS {part14ObjectClass ??};

A.2.8 Test Infrastructure Test Object

testInfrastructureTestObject MANAGED OBJECT CLASS
 DERIVED FROM "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject;
 CHARACTERIZED BY

testInfrastructureTestPkg PACKAGE

BEHAVIOUR testInfrastructureTestBehaviour BEHAVIOUR

DEFINED AS "The test infrastructure test is a 'null test' whose purpose is solely to investigate the ability of a managed open system to behave as specified in ITU-T Rec. X.745 |

ISO/IEC 10164-12 – that is, to accept an incoming test request and generate appropriate responses. Not test exercises are performed on resources within or external to the managed open system. If a controlled test is requested, an instance of this class will be created which will exist for the time taken to progress through the test cases defined by ITU-T Rec. X.745 | ISO/IEC 10164-12. During this time, a test conductor may issue control messages (suspend, resume, terminate) to the instance. The time taken to progress through the states can be longer than the timeout value specified thereby deliberately causing the test outcome to be TIMEOUT and the instance to be deleted.";;;

CONDITIONAL PACKAGES

```
testInfrastructureTestResultPackage PACKAGE NOTIFICATIONS
"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testResultNotification testInfrastructureControlledResultsParam;;
PRESENT IF !Unsolicited reporting is supported.!,
stateTransitionTimeIntervalPkg PRESENT IF !stateTransitionTimeInterval is present in the initAttributeList
parameter of the testRequestControlled request.!,
```

REGISTERED AS {part14ObjectClass ??};

A.3 Package definitions

connectivityThresholdPkg PACKAGE

BEHAVIOUR connectivityThresholdBehaviour BEHAVIOUR

DEFINED AS !The connectivity threshold is either defined explicitly in the test object or it is specified indirectly as an attribute in the MORT. It indicates the maximum acceptable error rate for the test. If the threshold is exceeded, the test is terminated and a fail outcome is returned.!;;

ATTRIBUTES testThreshold GET-REPLACE;

REGISTERED AS {part14Package ??};

DataIntegrityResultsPkg PACKAGE

BEHAVIOUR dataIntegrityResultsBehaviour BEHAVIOUR

DEFINED AS !This attribute contains the test results of a data integrity test.!;;

ATTRIBUTES dataIntegrityResults GET;

REGISTERED AS {part14Package ??};

dataIntegrityThresholdPkg PACKAGE

BEHAVIOUR dataIntegrityResultsBehaviour BEHAVIOUR

DEFINED AS !The testThreshold attribute identifies the maximum Timeout period for the data to be transferred to the associated object and reflected back to the MORT. If the data is returned within this period, the test will terminate indicating that the test outcome is pass. If this threshold is exceeded, it will cause the termination of the data integrity test and will indicate that the test outcome is fail.!;;

ATTRIBUTES testThreshold GET-REPLACE;

REGISTERED AS {part14Package ??};

dataUnitsPkg PACKAGE

BEHAVIOUR dataUnitsBehaviour BEHAVIOUR

DEFINED AS !The type and quantity of test data units to be sent is specified by the dataUnits attribute of the test. This attribute is only used if the test supports manager specification of the type and/or the quantity of data units.!;;

ATTRIBUTES dataUnits GET-REPLACE;

REGISTERED AS {part14Package ??};

loopbackResultsPkg PACKAGE

BEHAVIOUR loopbackResultsBehaviour BEHAVIOUR

DEFINED AS !The loopbackResults attribute contains the test results of a loopback test. The resultTestDuration must be supplied if this result is for a period shorter than the test pattern duration. The meaning of not sending resultTestDuration is that the period of time covered by this result is the whole test pattern duration. The patternType is used to notify the Test Conductor of the type of data this result applies to, and is the same value as the DataUnits attribute. The error cause should be set if the Test Performer is able to determine a reason for the error.!;;

ATTRIBUTES loopbackResults GET;

REGISTERED AS {part14Package ??};

loopbackThresholdPkg PACKAGE

BEHAVIOUR loopbackThresholdBehaviour BEHAVIOUR

DEFINED AS !The testThreshold attribute identifies an explicit error threshold which if crossed causes the termination of the loopback test.!;;
ATTRIBUTES testThreshold GET-REPLACE;

REGISTERED AS {part14Package 6};

loopbackTimeoutPkg PACKAGE

BEHAVIOUR loopbackTimeoutBehaviour BEHAVIOUR

DEFINED AS !The loopback timeout attribute contains the value for the timeout period which measures from the start of transmission of each test pattern to the start of the receipt of the corresponding pattern. When the loopback transmission delay time exceeds the loopbackTimeout period, the whole test is terminated and the loopback timeout period must be returned as part of the monitoredAttribute parameter of the testResultNotification and the test outcome is set to Fail. The loopbackTxDelayParam should be returned in all testResultNotifications where the loopback delay time is less than the loopbackTimeout period.!;;

ATTRIBUTES loopbackTimeout GET-REPLACE;
NOTIFICATIONS Rec. X.745 (1993) | ISO/IEC 10164-12:1994:
 testResultNotification loopbackTxDelayParam

REGISTERED AS {part14Package ??};

pDUReceptionPackage PACKAGE

BEHAVIOUR pDUReceptionBehaviour BEHAVIOUR

DEFINED AS !The pDUReception attribute, if used, indicates the conditions for protocol element transmission. When the protocol element received by the MORT matches the conditions specified in the pDUReception, the next protocol element is transmitted. The pDUReception attribute should not be used if the waitingInterval attribute is used.!;;

ATTRIBUTES pDUReception GET-REPLACE;

REGISTERED AS {part14Package ??};

protocolIntegrityResultsPackage PACKAGE

BEHAVIOUR protocolIntegrityResultsBehaviour;
ATTRIBUTES protocolIntegrityResults GET;

REGISTERED AS {part14Package ??};

resultIntervalPkg PACKAGE

BEHAVIOUR resultIntervalBehaviour BEHAVIOUR

DEFINED AS !A "zero" value means that there are no intermediate results. A value other than "zero" specifies the interval (in seconds) between the generation of intermediate results.!;;

ATTRIBUTES resultInterval GET-REPLACE;

REGISTERED AS {part14Package ??};

stateTransitionTimeIntervalPkg PACKAGE

BEHAVIOUR stateTransitionTimeIntervalBehaviour BEHAVIOUR

DEFINED AS !The stateTransitionTimeInterval attribute contains the time interval to be used to progress through the test states of a controlled test.!;;

ATTRIBUTES stateTransitionTimeInterval GET-REPLACE;

REGISTERED AS {part14Package ??};

testConditionsPkg PACKAGE

BEHAVIOUR testConditionsBehaviour BEHAVIOUR

DEFINED AS !The testConditions attribute specifies under what conditions the resources should be allocated to the test. In particular, it indicates whether the test should initiate if the MORT is busy and whether the user of the MORT can cause the test to abort.!;;

ATTRIBUTES testConditions GET-REPLACE;

REGISTERED AS {part14Package ??};

timeoutPeriodPkg PACKAGE

BEHAVIOUR timeoutPeriodBehaviour BEHAVIOUR

DEFINED AS !The timeoutPeriod attribute contains the value for the timeout period, measured from the initiation of the test. When the execution of a test exceeds the timeout period, the test is terminated and the timeout period is returned as part of the test result and the test outcome is set to TIMEOUT.!;;

ATTRIBUTES timeoutPeriod GET-REPLACE;

REGISTERED AS {part14Package ??};

waitingIntervalPackage PACKAGE
 BEHAVIOUR waitingIntervalBehaviour;
 ATTRIBUTES waitingInterval GET-REPLACE;

REGISTERED AS {part14Package ??};

A.4 Name binding definitions

testObject-System-NB NAME BINDING
 SUBORDINATE OBJECT CLASS
 "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObject AND SUBCLASSES;
 NAMED BY SUPERIOR OBJECT CLASS
 "Rec. X.721 (1992) | ISO/IEC 10165-2:1992":system AND SUBCLASSES;
 WITH ATTRIBUTE "Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testObjectId;
 DELETE;

REGISTERED AS {part14NameBinding 1};

A.5 Parameter definitions

A.5.1 Connection controlled results

connectionControlledResultsParam PARAMETER
 CONTEXT EVENT-INFO;
 ATTRIBUTE TestCategories-ASN1Module.ConnectionTestResults;
 BEHAVIOUR connectionResultsParamBehaviour BEHAVIOUR
 DEFINED AS !This parameter may be used to convey additional connection test result information that is related to the function being tested.!;;;

A.5.2 Connection test info

connectionTestInfoParam PARAMETER
 CONTEXT ACTION-INFO;
 WITH SYNTAX TestCategories-ASN.1Module.ConnectionTestInfo;
 BEHAVIOUR connectionTestInfoParamBehaviour;
 DEFINED AS !This parameter is used to convey additional connection test information that is related to the function being tested.!;;;

A.5.3 Connection uncontrolled results

connectionUncontrolledResultsParam PARAMETER
 CONTEXT ACTION-REPLY;
 WITH SYNTAX TestCategories-ASN1Module.ConnectionTestResults;
 BEHAVIOUR connectionUncontrolledResultsParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.4 Connectivity controlled results

connectivityControlledResultsParam PARAMETER
 CONTEXT EVENT-INFO;
 ATTRIBUTE connectivityResults;
 BEHAVIOUR connectivityResultsParamBehaviour
 DEFINED AS !This parameter may be used to convey additional connectivity test result information that is related to the function being tested.!;;;

A.5.5 Connectivity uncontrolled results

connectivityUncontrolledResultsParam PARAMETER
 CONTEXT ACTION-REPLY;
 WITH SYNTAX TestCategories-ASN1Module.ConnectivityTestResults;
 BEHAVIOUR connectivityUncontrolledResultsParamBehaviour;
 DEFINED AS !This parameter is used to convey connection test result information that is related to the function being tested.!;;;

REGISTERED AS {part14Parameter ??};

A.5.6 Connectivity test info

connectivityTestInfoParam PARAMETER
CONTEXT ACTION-INFO;
WITH SYNTAX TestCategories-ASN1Module.ConnectivityTestInfo;
BEHAVIOUR connectivityTestInfoParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.7 Data integrity controlled results

dataIntegrityControlledResultsParam PARAMETER
CONTEXT EVENT-INFO;
ATTRIBUTE dataIntegrityResults;
BEHAVIOUR dataIntegrityControlledResultsParamBehaviour;;

A.5.8 Data integrity uncontrolled results

dataIntegrityUncontrolledResultsParam PARAMETER
CONTEXT ACTION-REPLY;
WITH SYNTAX TestCategories-ASN1Module.DataIntegrityTestResults;
BEHAVIOUR dataIntegrityUncontrolledResultsParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.9 Data integrity test info

dataIntegrityTestInfoParam PARAMETER
CONTEXT ACTION-INFO;
WITH SYNTAX TestCategories-ASN1Module.DataIntegrityTestInfo;
BEHAVIOUR dataIntegrityTestInfoParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.10 Loopback controlled results

loopbackControlledResultsParam PARAMETER
CONTEXT EVENT-INFO;
ATTRIBUTE TestCategories-ASN1Module.LoopbackTestResults;
BEHAVIOUR loopbackControlledResultsParamBehaviour;;

A.5.11 Loopback uncontrolled results

loopbackUncontrolledResultsParam PARAMETER
CONTEXT ACTION-REPLY;
WITH SYNTAX TestCategories-ASN1Module.LoopbackTestResults;
BEHAVIOUR loopbackUncontrolledResultsParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.12 Loopback test info

loopbackTestInfoParam PARAMETER
CONTEXT ACTION-INFO;
WITH SYNTAX TestCategories-ASN1Module.LoopbackTestInfo;
BEHAVIOUR loopbackTestInfoParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.13 Protocol integrity controlled results

protocolIntegrityControlledResultsParam PARAMETER
CONTEXT EVENT-INFO;
ATTRIBUTE TestCategories-ASN1Module.ProtocolIntegrityTestResults;
BEHAVIOUR protocolIntegrityControlledResultsParamBehaviour;;

A.5.14 Protocol integrity uncontrolled results

protocolIntegrityUncontrolledResultsParam PARAMETER
 CONTEXT ACTION-REPLY;
 WITH SYNTAX TestCategories-ASN1Module.ProtocolIntegrityTestResults;
 BEHAVIOUR protocolIntegrityUncontrolledResultsParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.15 Protocol integrity test info

protocolIntegrityTestInfoParam PARAMETER
 CONTEXT ACTION-INFO;
 WITH SYNTAX TestCategories-ASN1Module.ProtocolIntegrityTestInfo;
 BEHAVIOUR protocolIntegrityTestInfoParamBehaviour;

REGISTERED AS {part14Parameter ??};

A.5.16 Resource boundary controlled results

resourceBoundaryControlledResultsParam PARAMETER
 CONTEXT EVENT-INFO;
 ATTRIBUTE TestCategories-ASN1Module.ResourceBoundaryTestResults;
 BEHAVIOUR resourceBoundaryControlledResultsParamBehaviour BEHAVIOUR
 DEFINED AS !The resourceBoundaryControlledResultsParam parameter contains the Resource Boundary intermediate test results.!;;

A.5.17 Resource boundary test info

resourceBoundaryTestInfoParam PARAMETER
 CONTEXT ACTION-INFO;
 WITH SYNTAX TestCategories-ASN1Module.ResourceBoundaryTestInfo;
 BEHAVIOUR resourceBoundaryTestInfoParamBehaviour BEHAVIOUR
 DEFINED AS !The resourceBoundaryTestInfoParam parameter contains the information of the Resource Boundary test, that may be filled in for the test. This is the initial content of the ResultReportIndicator and SequenceOfEvents attributes.!;;

REGISTERED AS {part14Parameter ??};

A.5.18 Resource self test info

resourceSelfTestInfoParam PARAMETER
 CONTEXT ACTION-INFO;
 WITH SYNTAX TestCategories-ASN1Module.ResourceSelfTestInfo;
 BEHAVIOUR resourceSelfTestInfoParamBehaviour;
 DEFINED AS !This parameter is used to convey additional resource self test information that is related to the function being tested.!;;

REGISTERED AS {part14Parameter ??};

A.6 Attribute definitions**A.6.1 Connection test results**

connectionTestResults ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ConnectionTestResults;
 MATCHES FOR EQUALITY;

REGISTERED AS {part14Attribute ??};

A.6.2 Connectivity results

connectivityResults ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ConnectivityTestResults;
 MATCHES FOR EQUALITY;

REGISTERED AS {part14Attribute ??};

A.6.3 Connectivity type

connectivityType ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.Type;
MATCHES FOR EQUALITY;
BEHAVIOUR connectivityTypeBehaviour BEHAVIOUR
DEFINED AS !The connectivity attribute provides information to identify whether the test is connection oriented and is tested by establishing a connection or connectionless and is tested by a protocol exchange.!;;
REGISTERED AS {part14Attribute ??};

A.6.4 Data integrity results

dataIntegrityResults ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.DataIntegrityTestResults;
MATCHES FOR EQUALITY;
REGISTERED AS {part14Attribute ??};

A.6.5 Data integrity type

dataIntegrityType ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.Type;
MATCHES FOR EQUALITY;
REGISTERED AS {part14Attribute ??};

A.6.6 Data units

dataUnits ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.DataUnits;
MATCHES FOR EQUALITY;
REGISTERED AS {part14Attribute ??};

A.6.7 End connection test results

endConnectionTestResults ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.EndConnectionTestResults;
MATCHES FOR EQUALITY;
REGISTERED AS {part14Attribute ??};

A.6.8 Error ratio report type

errorRatioReportType ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ErrorRatioReportType;
MATCHES FOR EQUALITY;
REGISTERED AS {part14Attribute ??};

A.6.9 Establishment time

establishmentTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.EstablishmentTime;
MATCHES FOR EQUALITY, ORDERING;
REGISTERED AS {part14Attribute ??};

A.6.10 Loopback results

loopbackResults ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.LoopbackTestResults;
REGISTERED AS {part14Attribute ??};

A.6.11 Loopback timeout

loopbackTimeout ATTRIBUTE
 WITH ATTRIBUTE SYNTAX ??????;
 MATCHES FOR EQUALITY;
 REGISTERED AS {part14Attribute ??};

A.6.12 Loopback type

loopbackType ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.LoopbackType;
 REGISTERED AS {part14Attribute ??};

A.6.13 PDU sequence

pDUSequence ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.PDUSequence;
 BEHAVIOUR pDUSequenceBehaviour BEHAVIOUR
 DEFINED AS !The pDUSequence specifies a sequence of protocol elements for transmission to the associated object during the protocol integrity test.!.
 REGISTERED AS {part14Attribute ??};

A.6.14 PDU reception

pDUReception ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.PDUReception;
 BEHAVIOUR pDUReceptionBehaviour;
 REGISTERED AS {part14Attribute ??};

A.6.15 Protocol integrity results

protocolIntegrityResults ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ProtocolIntegrityTestResults;
 REGISTERED AS {part14Attribute ??};

A.6.16 Resource Boundary Test Results

resourceBoundaryTestResults ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ResourceBoundaryTestResults;
 BEHAVIOUR resourceBoundaryTestResultsBehaviour BEHAVIOUR
 DEFINED AS !The resource boundary test results attribute contains the intermediate results of the test. The information in the attribute depends on the event that triggers the reporting. There are four cases that trigger the reporting of intermediate test results:

- Case 1 – Completion of last signal in sequence of test events (if indicated by the Result Report Indicator).
- Case 2 – Receipt of a signal at a PCO while no event sequence list is active and therefore no receive signal is to be matched.
- Case 3 – Receipt of a signal at a PCO while another receive signal is to be matched according to the active event sequence list.
- Case 4 – Wait duration timer expired in the case of a receive signal.

For these cases, the intermediate test results are as follows:

In case 1:

- Intermediate Resource Boundary Test outcome (passed)

In case 2:

- Intermediate Resource Boundary Test outcome (unexpected)
- Signal Type of the received signal
- Signal values of the signal type
- Identities of the MORTs from which the signal was received
- Identities of the AOs where the signal was received

In case 3:

- Intermediate Resource Boundary Test outcome (wrongSignalReceived)
- Event Id of failed test event
- Signal Type of the received signal
- Signal values of the signal type
- Identities of the AOs where the signal was received

In case 4:

- Intermediate Resource Boundary Test outcome (waitDurationTimerExpired).

This attribute contains either the details of a failed test and/or the information on the received signal at a PCO. It contains information on a received signal if a signal was received that was not specified as expected in a sequence of events. It contains the details of a failure in the case of a failed test. Two kind of failures may be reported:

- A wrong signal was received: another signal was received than the signal as specified in the sequence of events.
- The wait duration timer expired: an expected signal was specified, but no signal was received.

In both cases, also the event identifier of the failed test event is reported.!

REGISTERED AS {part14Attribute ??};

A.6.17 Resource self test results

resourceSelfTestResults ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ResourceSelfTestResults;

REGISTERED AS {part14Attribute ??};

A.6.18 Result interval

resultInterval ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ResultInterval;
MATCHES FOR EQUALITY, ORDERING;

REGISTERED AS {part14Attribute ??};

A.6.19 Result Report Indicator

resultReportIndicator ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.ResultReportIndicator;
BEHAVIOUR resultReportIndicatorBehaviour BEHAVIOUR
DEFINED AS !The resultReportIndicator attribute indicates whether a result report is required when a sequence of test events was passed.!!;

REGISTERED AS {part14Attribute ??};

A.6.20 Sequence of Events

sequenceOfEvents ATTRIBUTE
WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.SequenceOfEvents;
BEHAVIOUR sequenceOfEventsBehaviour BEHAVIOUR
DEFINED AS !The sequenceOfEvents attribute specifies a sequence of signals that have to be inserted or received at the specified MORT and AO. In the case of a send signal, it is inserted in the MORT via the specified AO. In the case of a receive signal, it is received at the indicated AO from the indicated MORT. Each event is numbered by an event identifier.

The sequence of test events may contain a single signal that has to be inserted or received, but also a sequence of events may be specified that consists of send and receive signals. The test performer shall then handle the events sequentially. In the case of a receive event, the test performer shall wait for the receipt of the indicated signal before continuing with the next event. It waits until the wait duration timer has expired. If expired, a failure occurred. In the case of a send signal, the wait duration timer specifies the time that is waited upon before the signal is inserted after the former event was completed. In the case of a failure the sequence of events is not completed, but is aborted.!

REGISTERED AS {part14Attribute ??};

A.6.21 State transition time interval

statefTransitionTimeInterval ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.StatefTransitionTimeInterval;
 MATCHES FOR EQUALITY;
 REGISTERED AS {part14Attribute ??};

A.6.22 Test conditions

testConditions ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.TestConditions
 MATCHES FOR EQUALITY;
 REGISTERED AS {part14Attribute ??};

A.6.23 Test pattern

testPattern ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.TestPattern;
 REGISTERED AS {part14Attribute ??};

A.6.24 Test patterns

testPatterns ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.TestPatternSequence;
 REGISTERED AS {part14Attribute ??};

A.6.25 Test threshold

testThreshold ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.TestThreshold;
 MATCHES FOR EQUALITY;
 REGISTERED AS {part14Attribute ??};

A.6.26 Waiting interval

waitingInterval ATTRIBUTE
 WITH ATTRIBUTE SYNTAX TestCategories-ASN1Module.WaitingInterval;
 MATCHES FOR EQUALITY, ORDERING;
 REGISTERED AS {part14Attribute ??};

A.7 Supporting productions

This subclause specifies the required ASN.1 value notation for the value reference used in the MANAGED OBJECT CLASS template.

TestCategories-ASN1Module { joint-iso-ccitt ms(9) function(2) part14(14) asn1Module(2) 1 }

DEFINITION IMPLICIT TAGS

::= BEGIN

```
--      EXPORTS everything
      IMPORTS      AttributeId
      FROM CMIP-1 { joint-iso-ccitt ms(9) cmip(1) modules0(0) protocol(3) }
      ManagementExtension, StartTime, StopTime
      FROM      Attribute-ASN1Module { joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1 }
      TimeoutPeriod, TestOutcome, Timespec, MORTs, AssociatedObjects
      FROM Test-ASN1Module { joint-iso-ccitt ms(9) function(2) part12(12) asn1Module(2) 0 };
```

```
part14ObjectClass      OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part14(14) managedObjectClass(3) }
part14Package          OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part14(14) package(4) }
part14Parameter       OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part14(14) parameter(5) }
part14NameBinding     OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part14(14) nameBinding(6) }
part14Attribute       OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part14(14) attribute(7) }
```

```

ConnectionTestInfo ::= SEQUENCE {
    testPattern [0] TestPattern OPTIONAL, -- test pattern sent
    testDirection [1] TestDirection OPTIONAL,
    testDuration [2] TestDuration OPTIONAL,
    reportingInterval [3] TimeSpec OPTIONAL }

ConnectionTestResults ::= SEQUENCE {
    receivedTestPattern [0] TestPattern OPTIONAL,
    errorRatio [1] Real OPTIONAL,
    testDirection [2] TestDirection OPTIONAL,
    testDurationMade [3] Timespec OPTIONAL }

ConnectivityTestInfo ::= SEQUENCE {
    timeoutPeriod [0] CHOICE {
        timeUnits Timespec,
        attributeId AttributeId },
    effectiveTime [1] Timespec }

ConnectivityTestResults ::= CHOICE {
    establishmentTime [0] TimeSpec,
    timeoutPeriod [1] TimeoutPeriod,
    testThreshold [2] TestThreshold,
    specificError [3] OBJECT IDENTIFIER }

ConnectivityThreshold ::= CHOICE {
    time [0] SEQUENCE {
        unitsType UnitsType,
        unitsTotal UnitsTotal },
    rawData [1] RawData }

ContCond ::= CHOICE {
    intervalTime Timespec,
    pduReception PDUREception }

DataCategory ::= INTEGER {
    bits (0),
    octets (1),
    blocks (2),
    packets (3)
    -- ... }

DataIntegrityTestInfo ::= SEQUENCE OF {
    dataUnits DataUnits }

DataIntegrityTestResults ::= CHOICE {
    testResult [0] SEQUENCE {
        establishmentTime [1] EstablishmentTime OPTIONAL,
        originalData [2] DataUnits,
        corruptedData [3] DataUnits OPTIONAL },
    timeoutPeriod [4] TimeoutPeriod,
    testThreshold [5] TestThreshold
    }

DataRate ::= CHOICE { REAL, OBJECT IDENTIFIER }

DataSize ::= INTEGER

DataType ::= CHOICE {
    integerDataType IntegerDataType,
    objectIdentifierDataType OBJECT IDENTIFIER }

DataUnits ::= SEQUENCE {
    dataType [0] DataType,
    dataCategory [1] DataCategory OPTIONAL,
    dataSize [2] DataSize OPTIONAL,
    dataNumber [3] UnitsTotal OPTIONAL,
    dataRate [4] DataRate OPTIONAL }

```

```

ErrorRatioReportType ::= ENUMERATED {
    errorBitNumber (0),
    percentErrorSecond (1) }

ErrorUnitThreshold ::= SEQUENCE {
    unitType      [0] UnitType,
    unitsTotal    [1] UnitsTotal }

EstablishmentTime ::= Timespec

FailedCase ::= ENUMERATED {
    lineDisconnected(0),
    counterOverflow(1) }          -- error counter overflow

IntegerDataType ::= INTEGER {
    allBitOn(0),
    allBitOff(1),
    incrementNumber(2),
    pn11(3), -- CCITT 0.152
    pn15(4), -- CCITT 0.151
    pn20(5)
    -- ...}

IntermediateResourceBoundaryTestOutcome ::= ENUMERATED {
    passed                (0),
    unexpected            (1),
    wrongSignalReceived  (2),
    waitDurationTimerExpired (3) }

IntermediateResponse ::= ENUMERATED {
    inProgress                (0),
    delayedMeasurement        (1),
    interruptedMeasurement    (2),
    repeatLater               (3),
    noAcknowledgement (4) } -- abnormal condition, a correct response cannot be sent

InternalResourceTestResults ::= SEQUENCE {
    functionTested          OBJECT IDENTIFIER,
    testResult              ANY DEFINED BY functionTested }

LoopbackError ::= CHOICE { errorBitNo  INTEGER, percent  REAL }

LoopbackTestInfo ::= SEQUENCE OF {
    loopbackData  TestSignalSequence  OPTIONAL,
    testStartTime GeneralizedTime      OPTIONAL,
    testIntervalTime  TimeSpec         OPTIONAL,
    reportingIntervalTime  TimeSpec    OPTIONAL,
    loopbackType  OBJECT IDENTIFIER    OPTIONAL,
    loopbackTimeout  TimeSpec          OPTIONAL,
    loopbackErrorThreshold  LoopbackError  OPTIONAL }

LoopbackTestResult ::= CHOICE {
    passed  SEQUENCE
        { loopbackDataReceived  TestSignalSequence,
          loopbackErrorReceived LoopbackError },
    fail  INTEGER (errorRatioThresholdExceeded (0), loopbackTimeoutExceeded (1)),
    timeout  NULL, -- test timeout period exceeded --
    prematureTermination  NULL,
    inconclusive  NULL }

LoopbackType ::= OBJECT IDENTIFIER

Parameter ::= SEQUENCE {
    attributeType  AttributeId,
    value          ANY DEFINED BY attributeType }

Pattern Type ::= DataType

```

PDUReception ::= SEQUENCE {
 pDUType **PDUType,**
 parameter **Parameter,**
 responseTimeout **ResponseTimeout** **OPTIONAL }**

PDUSequence ::= SEQUENCE {
 pDUType **PDUType,**
 parameter **Parameter,**
 contCond **ContCond,**
 waitDuration **WaitDuration** **OPTIONAL}**

PDUType ::= OBJECT IDENTIFIER

ProtocolIntegrityTestInfo ::= SEQUENCE { *-- This ProtocolIntegrityTestInfo is optional.*
 pDUSequence **[0] PDUSequence,**
 pDUReception **[1] PDUReception** **OPTIONAL,** *-- one, and only one, of the pDUReception*
 waitingInterval **[2] WaitingInterval** **OPTIONAL,** *-- and the waitingInterval shall be present*
 startTime **[3] StartTime** **OPTIONAL,**
 stopTime **[4] StopTime** **OPTIONAL }**

ProtocolIntegrityTestResults ::= INTEGER {
 wrongPDUResponse **(0),**
 pDUResponseNotReceived **(1) }**

ResourceBoundaryTestInfo ::= SEQUENCE {
 resultReportIndicator **[0] IMPLICIT ResultReportIndicator** **OPTIONAL,**
 sequenceOfEvents **[1] IMPLICIT SequenceOfEvents** **OPTIONAL }**

ResourceBoundaryTestResults ::= SEQUENCE {
 signalReceived **[0] IMPLICIT SignalReceived** **OPTIONAL,**
 intermediateResourceBoundaryTestOutcome **[1] IMPLICIT IntermediateResourceBoundaryTestOutcome**
 OPTIONAL,
 eventId **[2] IMPLICIT INTEGER** **OPTIONAL }**

ResourceSelfTestInfo ::= SEQUENCE {
 diagnosticType **[0] INTEGER** **OPTIONAL,** *--type of diagnostic to be run*
 phases **SET OF INTEGER,** *-- phases to be run*
 iteration **[1] INTEGER** *-- number of iterations for each type*
 timeoutPeriod **[2] TimeoutPeriod }**

ResourceSelfTestResults ::= CHOICE {
 intermediateResponse **IntermediateResponse,**
 SEQUENCE {
 phases **SET OF INTEGER,**
 iteration **[0] INTEGER,**
 timeoutPeriod **[1] TimeoutPeriod,**
 finalResponse **PrintableString** **OPTIONAL }** **}**

ResultInterval ::= INTEGER

ResultReportIndicator ::= ENUMERATED {
 resultReportForPassedSequences **(0),**
 noResultReportForPassedSequences **(1) }**

SequenceOfEvents ::= SET OF SEQUENCE {
 eventId **INTEGER,**
 signalType **OBJECT IDENTIFIER,**
 signalValue **ANY DEFINED BY signalType,**
 signalDirection **SignalDirection,**
 mORTs **MORTs,**
 associatedObjects **AssociatedObjects,**
 waitDuration **WaitDuration }**

SignalDirection ::= ENUMERATED {
 send **(0),**
 receive **(1) }**

```

SignalReceived ::= SET OF SEQUENCE {
    signalType      OBJECT IDENTIFIER,
    signalValue     ANY DEFINED BY signalType,
    mORTs          MORTs,
    associatedObjects AssociatedObjects }

StateTransitionTimeInterval ::= Timespec

TerminationReason ::= OBJECT IDENTIFIER

TestConditions ::= SEQUENCE {
    INTEGER {
        testIfBusy      (0),
        rejectIfBusy    (1),
        waitIfBusy      (2) },
    INTEGER {
        customerOverrideTest (0),
        noCustomerOverrideTest (1) } OPTIONAL }

TestDirection ::= INTEGER { atoZ (0), ztoA (1), bothDirections (2), transmitFromNearEnd (3), receiveFromNearEnd (4) }

TestDuration ::= CHOICE {
    signalDuration Timespec,
    signalLength SEQUENCE {
        size INTEGER
        dataRate dataRate OPTIONAL }

TestPatternSequence ::= SEQUENCE OF { TestSignal }-- There should be some size limitation here
-- if test signal pattern is to fit in one pdu!!!!

TestPattern ::= CHOICE {
    rawData OCTET STRING,
    standardType DataType,
    undefinedType ManagementExtension }

TestThreshold ::= CHOICE {
    bitErrorThreshold [0] INTEGER,
    percentageErrorThreshold [1] REAL,
    errorUnitThreshold [2] UnitErrorThreshold }

Type ::= INTEGER {
    connectionOriented (0),
    connectionless (1) }

UnitsTotal ::= INTEGER

UnitType ::= OBJECT IDENTIFIER

WaitDuration ::= Timespec

WaitingInterval ::= Timespec

END

```

Annexe B

Formulaire MCS

(Ce formulaire sera publié sous forme d'amendement à la présente Recommandation | Norme internationale)

Annexe C

Formulaire MICS

(Ce formulaire sera publié sous forme d'amendement à la présente Recommandation | Norme internationale)

Annexe D

Formulaire MOCS

(Ce formulaire sera publié sous forme d'amendement à la présente Recommandation | Norme internationale)

Annexe E

Formulaire MRCS

(Ce formulaire sera publié sous forme d'amendement à la présente Recommandation | Norme internationale)

Annexe F

Récapitulatif des catégories de tests

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

This annex provides, in the form of a matrix, a summary of all the test categories, identifying for each test category the TO attributes used, the use made of Test Info parameters and the possible Test Results.

Test category	Test category description	TO attributes	Test info	Test results
Connection	Ability of a comm path (real or virtual – MORT) to support a desired service with level of functionality, test = send test pattern over service in one direction	MORTs, AOs, testPatterns, connTestResults (Null for pass or testPattern and specError for fail), adminSt, timeoutPeriod, (test patterns twice, once as attribute, once in results)	–	Null for pass or testPattern and specError for fail (doesn't indicate if test pattern sent or received)
Connectivity	Verifies that connectivity may be established between two entities (MORT and AO) within specified time: for CO – establish conn, CL – exchange data units	MORTs, AOs, connectivityType, connectivityResults, establishment time (2X), timeout period	timeoutPeriod (timeUnits or attId), effectiveTime	CHOICE: estabTime, timeout Period, testThreshold, specificError
Data integrity	Whether two entities (MORT and AO) can exchange data without any corruption and measure time taken for exchange (test = sendPDU, get it back)	MORTs, AOs, actualTestTimePkg, dataIntegrityType, test patterns, timeoutPeriod, dataIntegrityResults, test threshold	DataUnits	CHOICE: testResult (estabTime, origData, corruptData), timeout Period, testThreshold (similar to test results for CY)
Loopback	Verifies that data can be sent and received, over a comm path within specific interval of time with acceptable rate (MORT is comm path – includes end pts) (test = send test pattern)	MORTs, AOs, loopbackType, testPatterns, dataUnit(s), errorRatioReportType, timeoutPeriod, testConditions, dataUnits, resultInterval, loopbackResults, loopbackThreshold	loopbackData, testStartTime, testTimeDuration, errorThresh, loopbackReportInterval, loopbackTestResultType (errorRatioReportType)	testTimeDuration, dataUnits, errorRatio, errorCause
Protocol integrity	Determines if MORT can conduct proper protocol interactions with specified AO	actualTesttime (actualStartTime, actualStopTime), PDU sequence, timeoutPeriod, PIResults, waitingInterval, pdu reception, test conditions	pduSeq, pduRec, waitingInterval, startTime, stopTime	wrongPDU response, pudResponse Not Received
Resource boundary	Multi-resource test using disconnected info streams; Verifies correctness of resources (MORT) internal to system that have points of control and observation (AOs)	AOs, resultReportIndicator, seqOfEvents (includes signal type (ANY)), RBResults, timeoutPeriod, testConditions	resultReportIndicator, seqOfEvents (signal type)	signalReceived, intermRBTestOutcome, eventId
Resource self	Ability of a resource (MORT, physical or logical function) to perform its allotted function	MORTs, timeoutPeriod, testConditions	–	–
Test infra-structure	Testing the test system; (same as testing TARR functionality)	–	–	–

Annexe G

Exemple de paquetage TARR pour le test de connexion

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

This annex provides an example of a TARR package for the connection test. A managed object containing such a package would support receiving a connection test request in the form of either a testRequestControlledAction action or a testRequestUncontrolledAction action with the appropriate parameters defined for connection test. A managed object which supports such a package is said to have "TARR functionality".

connectionTestTARRPackage PACKAGE

BEHAVIOUR "A managed object containing this package supports receiving a connection test request in the form of either a testRequestControlledAction action or a testRequestUncontrolledAction action with the appropriate parameters defined for connection test. It will return the results of the test in a testResultNofication with the connectionControlledResultsParam for a controlled test and in the action reply for the testRequestUncontrolledAction with the connectionUncontrolledResultsParam.";

ACTIONS

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testRequestControlledAction
connectionTestInfoParam

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectNotAvailable

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":independentTestInvocationError

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mistypedTestCategoryId

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mistypedTestRequestInformation

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTNotAvailable

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchAssociatedObject

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchMORT

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":relatedTOError,

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testRequestUncontrolledAction
connectionTestInfoParam

connectionUncontrolledResultsParam

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":associatedObjectNotAvailable

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mistypedTestCategoryId

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mistypedTestRequestInformation

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":mORTNotAvailable

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchAssociatedObject

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchMORT,

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testSuspendResumeAction

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchTestInvocationId

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchTestSessionId

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":invalidTestOperation

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testSuspendResumeError,

ISO/CEI 10164-14 : 1996 (F)

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testTerminateAction

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchTestInvocationId

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":noSuchTestSessionId

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":invalidTestOperation

"Rec. X.745 (1993) | ISO/IEC 10164-12:1994":testTerminateError;

REGISTERED AS {...};

Annexe H

Catégorie de test de limite de ressource

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

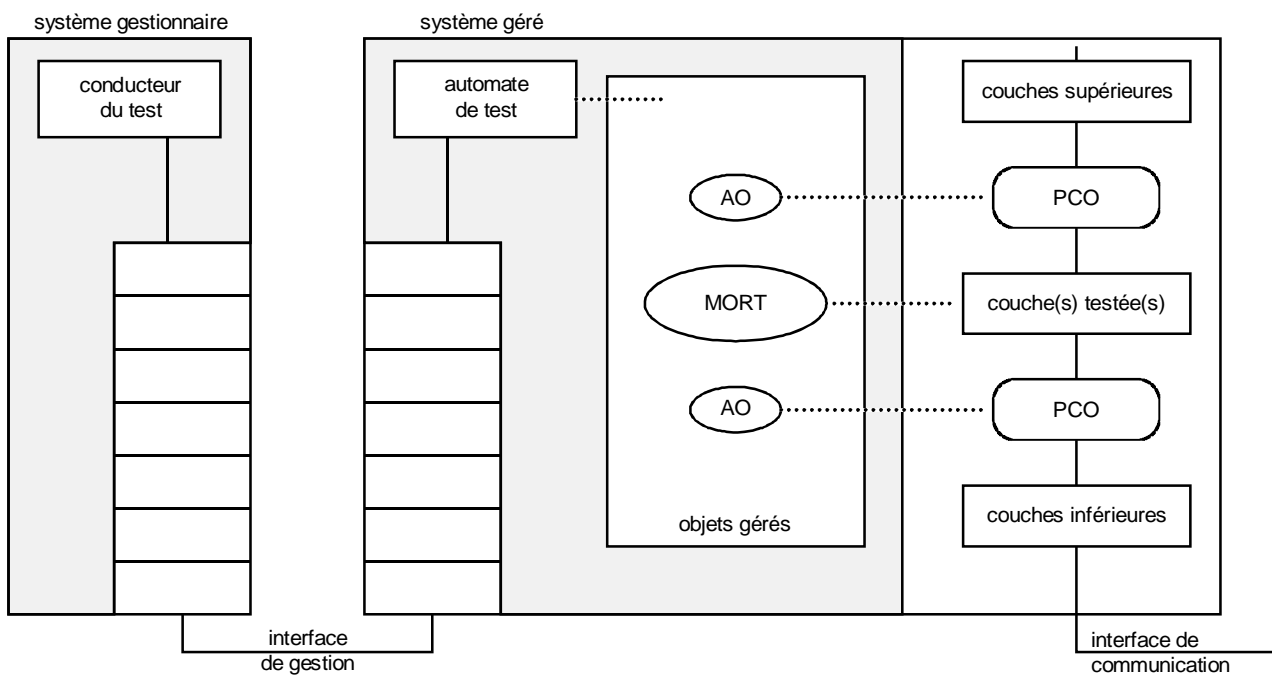
H.1 Le test de limite de ressource offre à l'utilisateur le moyen de tester certaines parties d'un système séparément comme si elles n'étaient pas encapsulées dans le système. Ce type de test est particulièrement approprié aux ressources ayant un comportement complexe qui ne peut être (entièrement) testé par l'observation des interfaces de communication externes.

Cette catégorie de test convient à plusieurs applications. Deux applications types, décrites ci-dessous, sont la vérification de réalisations de protocoles et l'analyse en limite de matériel.

H.1.1 Vérification des réalisations de protocoles

L'efficacité des tests de protocole dépend largement de l'accessibilité du protocole. Lors de l'utilisation du test de limite de ressource, les couches de protocole peuvent être testées indépendamment les unes des autres, procédure définie en tant qu'architecture de test conceptuelle dans l'ISO 9646-1. La Figure H.1 montre la configuration appliquée.

Les ressources de la catégorie de test représentent les couches de protocole, les points PCO représentent les points d'accès au service (SAP) entre les couches, et les signaux représentent les unités de données de protocole (PDU) ainsi que les primitives de service (SP) échangées entre les couches de protocole.



T0722930-95/d18

Figure H.1 – Mises en œuvre de protocole de test

H.1.2 Commande de tests d'analyse en limite de matériel

L'équipement numérique contenant des dispositifs pour les tests d'analyse de limite des circuits intégrés peut être commandé à partir d'un système de gestion tel qu'il est défini dans la notice IEEE 1149.1. La Figure H.2 montre la configuration appliquée.

Les ressources de la catégorie de test représentent les circuits intégrés, les points PCO représentent les cellules d'analyse de limite et les signaux représentent les vecteurs de test.

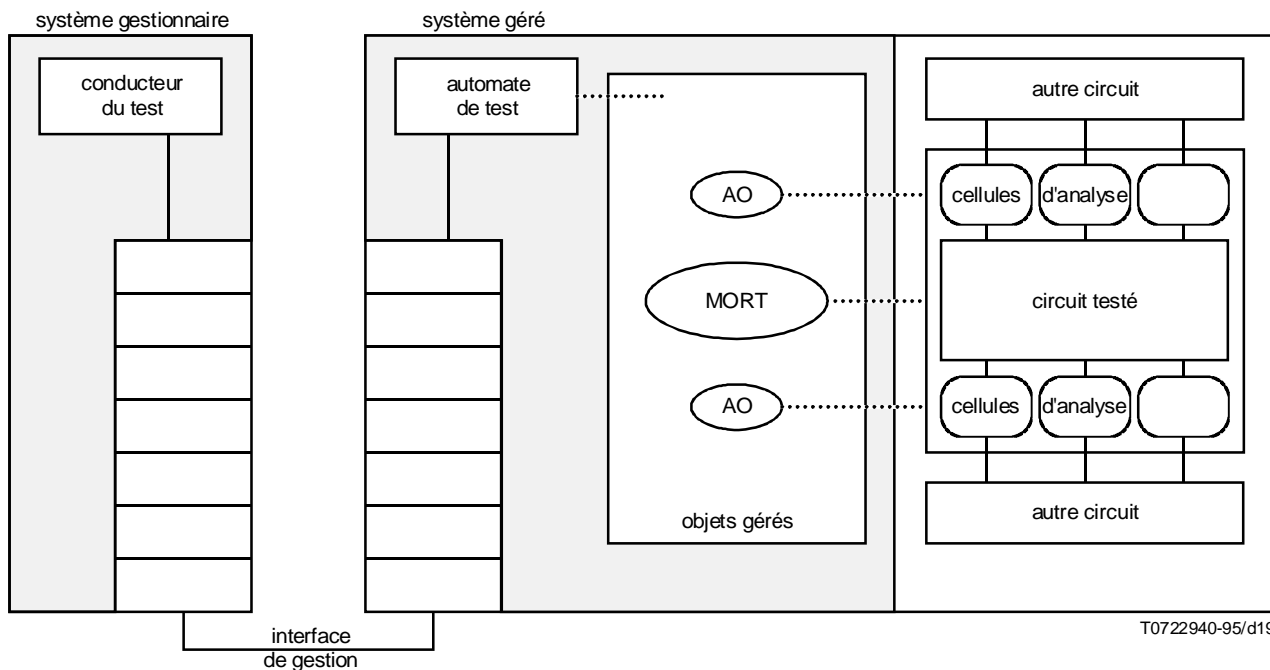


Figure H.2 – Test d'analyse de limite de matériel

Annexe I

Exemple de processus de test pour un réseau de communication géré

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Cette annexe illustre comment les catégories de test définies dans la présente Recommandation | Norme internationale pourraient être appliquées au problème de l'essai d'un réseau de systèmes de communication gérés.

La Figure I.1 montre un réseau à titre d'exemple. Ce réseau contient:

- un conducteur du test (partie du système ouvert de gestion) qui demande des tests et reçoit des comptes rendus de résultat conformément à la Rec. UIT-T X.745 | ISO/CEI 10164-12. A noter que ces comptes rendus de résultat peuvent être produits par tout système ouvert participant au test;
- un certain nombre de systèmes ouverts gérés contenant chacun un automate de test qui reçoit les demandes de test et déclenche les opérations internes nécessaires pour y répondre;
- des connexions qui transportent des informations entre les systèmes ouverts gérés;
- des connexions entre les systèmes ouverts qui contiennent des automates de test et des systèmes ouverts ou autres ne contenant pas d'automates de test.

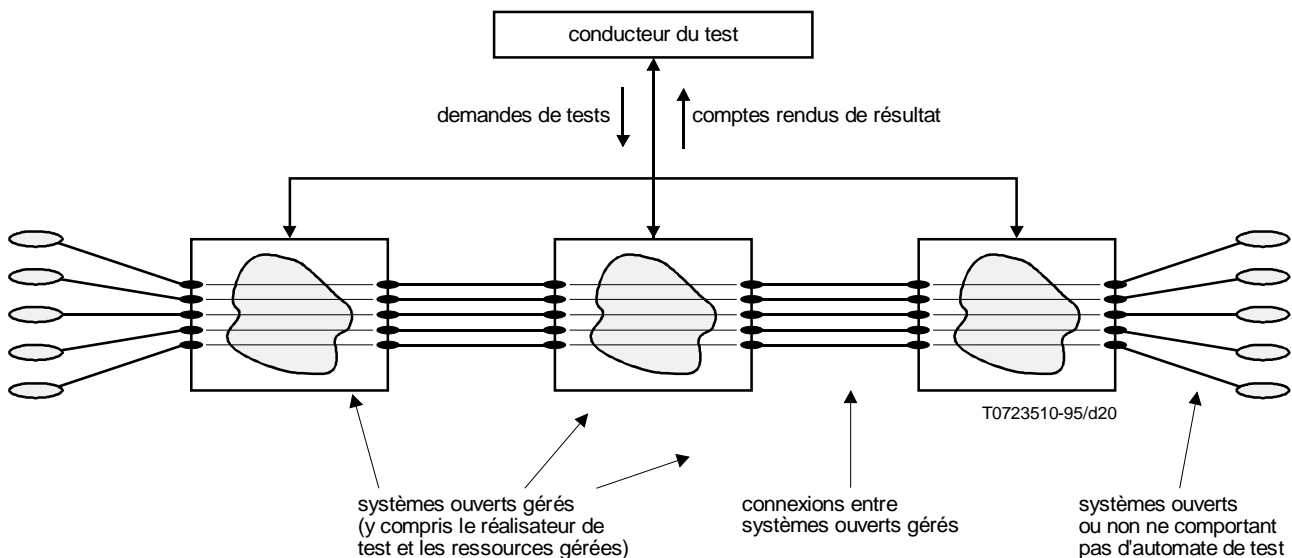


Figure I.1 – Réseau de systèmes de communication gérés

En ce qui concerne ce modèle de réseau, on peut définir un processus de test générique comportant cinq parties de base. Cependant, un processus de test réel peut ne pas comporter toutes ces parties (par exemple, il peut ne s'appliquer qu'à l'autotest de ressource).

1) *Test de l'interface avec les ressources du réseau*

Premièrement, le conducteur du test doit prouver aussi bien l'intégrité des trajets par lesquels passeront les demandes de test et les comptes rendus de résultat que l'aptitude des automates de test dans les systèmes ouverts gérés à recevoir des demandes et (s'il y a lieu) des commandes de test et à y répondre. On peut utiliser à cet effet le **Test d'infrastructure d'essai** défini dans la présente Recommandation | Norme internationale.

2) *Test des ressources du réseau*

Lorsqu'on a vérifié l'infrastructure d'essai, on peut utiliser cette infrastructure pour déclencher et observer des tests. Les **autotests de ressource** et les **tests de limite de ressource** permettent d'étudier la capacité de fonctionnement des ressources (composantes fonctionnelles ou physiques dans un système ouvert géré).

Les résultats de ces tests donneront une indication de l'état de fonctionnement de la ressource et, le cas échéant, des informations de diagnostic.

A noter que l'utilisation de l'**autotest de ressource** ne nécessite qu'une participation limitée de l'utilisateur car les caractéristiques détaillées des tests à exécuter sont intégrées dans la ressource à tester (objet MORT). L'utilisation du **test de limite de ressource** nécessite que l'utilisateur spécifie les trames de test ou les exercices à appliquer et, à titre facultatif, les résultats escomptés. Il est donc possible d'utiliser les **tests de limite de ressource** à la suite de la détection d'un défaut par un **autotest de ressource** afin d'obtenir des informations de diagnostic complémentaires.

3) *Test des connexions entre les ressources*

Dans l'hypothèse où les diverses ressources gérées sont exemptes de défaut (ou se sont révélées exemptes de défaut à la suite de l'exécution d'autotests de ressource), on peut les utiliser pour appliquer des tests aux diverses connexions entre les systèmes du réseau et recourir, à cet effet, aux tests suivants définis dans la présente Recommandation | Norme internationale:

- test de connexion;
- test de connectivité;
- test d'intégrité des données;
- test de bouclage;
- test d'intégrité de protocole.

Le **test de connexion** est comparable à l'**autotest de ressource** en ce sens que la nature du test exécuté est déterminée par le type de l'objet MORT et que les trames de test ou les exercices sont intégrés dans les objets AO participant au test. Seule une participation limitée de l'utilisateur est donc nécessaire. Pour s'assurer que les ressources gérées connectées peuvent coopérer à l'exécution de ces tests, un accord est nécessaire sur les caractéristiques détaillées des tests à exécuter. Par exemple, la séquence exacte des instructions ou des données transmises par la ressource émettrice doit être connue à l'avance par la ressource réceptrice afin qu'un résultat «succès» ou «échec» puisse être notifié. Les caractéristiques détaillées des tests de connexion point à point seront spécifiques du type de connexion considéré.

4) *Test des connexions entrant dans le réseau ou en sortant*

On peut également effectuer des tests sur les extrémités du réseau de communication géré, par exemple sur des connexions à l'entrée ou à la sortie d'équipements terminaux tels que des appareils téléphoniques et des modems.

Le **test de connexion** est un exemple de test que l'on pourrait utiliser pour répondre à cet objectif. Ce test est exécuté par une seule ressource gérée (objet AO) et ne nécessite, au maximum, qu'une participation limitée de l'équipement terminal distant. Par exemple, il peut être nécessaire de demander à l'utilisateur de déconnecter l'équipement terminal ou de mettre manuellement en place un bouclage avant le début du test. La ressource gérée qui exécute le test peut également commander l'établissement d'un bouclage en envoyant une tonalité ou une instruction définies.

On peut également utiliser à cet effet un **test de bouclage**.

5) *Test d'intégrité globale du réseau*

Jusqu'ici, seules certaines parties du réseau, c'est-à-dire les ressources constitutives et les interconnexions associées, ont été testées. Il est possible que, même si certaines parties fonctionnent correctement lorsqu'elles sont testées séparément, l'ensemble du réseau ne se comporte pas comme il le devrait (par exemple, en raison de la non-concordance de caractéristiques de fonctionnement, de problèmes de synchronisation, etc.). Il y a donc lieu de confirmer l'aptitude globale du réseau à assurer le service requis.

On peut, à cet effet, effectuer un test «d'intégrité du service» qui peut être considéré comme un **autotest de ressource** appliqué à l'ensemble du réseau géré (c'est-à-dire que le réseau est considéré, à un niveau supérieur d'abstraction, comme formant un seul système ouvert géré). La nature des tests effectués dépendra de la conception du réseau et devra donc être spécifiée dans le cadre des tâches relatives à la conception du réseau.

Le processus de test global est illustré sur la Figure I.2.

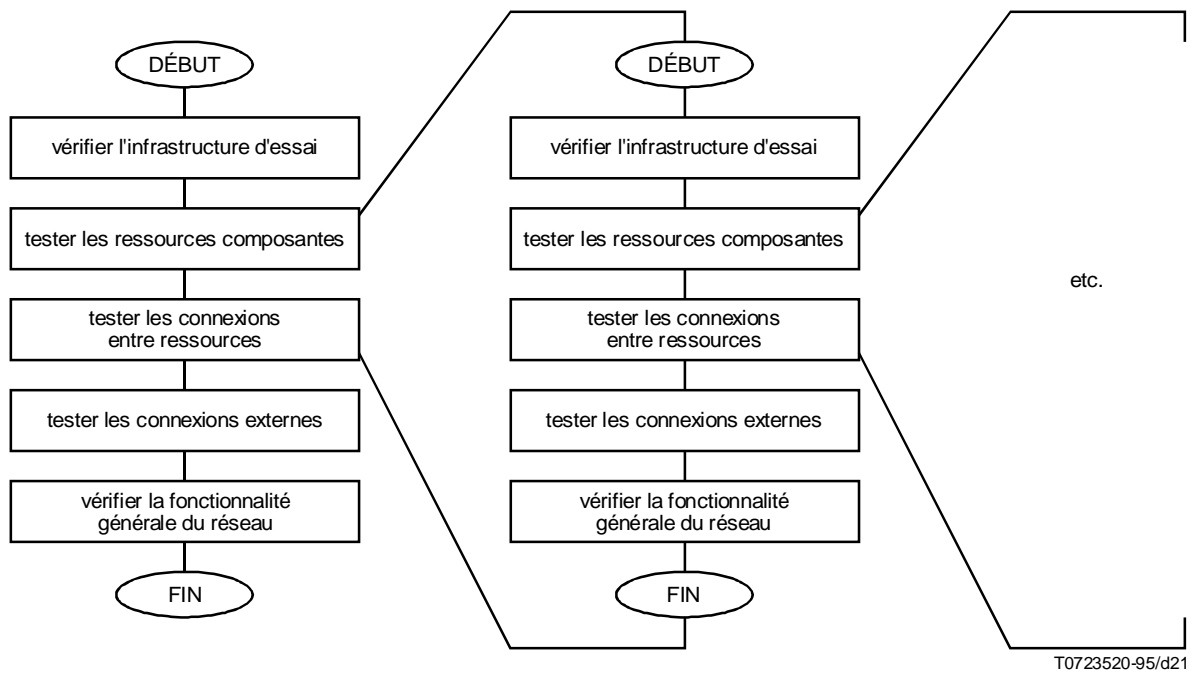


Figure I.2 – Processus de test générique

A noter que ce processus est hiérarchique. Un système ouvert géré peut être lui-même un réseau de systèmes ouverts gérés subordonnés, auquel cas il peut répondre à une demande d'exécution d'autotest de ressource ou de connexion en déclenchant un processus de test «fille» sous la forme décrite ci-dessus.

A noter également que, même si ce processus est indépendant des fonctions et de la mise en œuvre des systèmes ouverts gérés, il peut prouver la capacité de fonctionnement du réseau ou permettre de localiser un défaut dans un système ouvert géré particulier ou une connexion particulière. Ce processus est donc suffisant pour effectuer un diagnostic initial des défauts dans un réseau «multiconstructeur».