UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T H.248.1

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(05/2002)

SÉRIE H: SYSTÈMES AUDIOVISUELS 7ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Procédures de communication

Protocole de commande de passelle: version 2

Recommandation UIT-T H.248.1

RECOMMANDATIONS UIT-T DE LA SÉRIE H

SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	H.220–H.229
Aspects système	H.230–H.239
Procédures de communication	**H.240–H.259**
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
SYSTÈMES ET ÉQUIPEMENTS TERMINAUX POUR LES SERVICES AUDIOVISUELS	H.300–H.399
SERVICES COMPLÉMENTAIRES EN MULTIMÉDIA	H.450–H.499
PROCÉDURES DE MOBILITÉ ET DE COLLABORATION	
Aperçu général de la mobilité et de la collaboration, définitions, protocoles et procédures	H.500–H.509
Mobilité pour les systèmes et services multimédias de la série H	H.510–H.519
Applications et services de collaboration multimédia mobile	H.520–H.529
Sécurité pour les systèmes et services multimédias mobiles	H.530–H.539
Sécurité pour les applications et services de collaboration multimédia mobile	H.540–H.549
Procédures d'interfonctionnement de la mobilité	H.550–H.559
Procédures d'interfonctionnement de collaboration multimédia mobile	H.560–H.569

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

Recommandation UIT-T H.248.1

Protocole de commande de passerelle: version 2

Résumé

Aux fins d'une meilleure évolutivité, la présente Recommandation décompose en sous éléments fonctionnels la fonction de passerelle H.323 définie dans la Rec. UIT-T H.246. Elle spécifie les protocoles utilisés par ces éléments pour communiquer, ce qui permet aux implémentations de passerelles H.323 d'avoir une évolutivité élevée et qui incite à la multiplication de capacités, telles que les commutateurs SS7, de réseau à commutation de circuits (RCC). Ces sous éléments permettent également aux passerelles H.323 d'être constituées d'éléments issus de multiples vendeurs répartis sur de multiples plates-formes physiques. L'objet de la présente Recommandation est d'ajouter les capacités actuellement définies pour les systèmes H.323, en vue d'offrir de nouveaux moyens d'effectuer les opérations déjà assurées dans les systèmes H.323.

La présente Recommandation apporte plusieurs améliorations à la version 1 de la Rec. UIT-T H.248.1

– analyse des propriétés, des signaux, des événements et des statistiques distincts;

– traitement multiplex amélioré;

– topologie des flux;

– description améliorée des profils;

– modification de la capacité ServiceChange

NOTE – La présente Recommandation comporte la nouvelle numérotation de la Rec. UIT-T H.248, ainsi que les Annexes A à E et l'Appendice I de celle-ci.

Source

La Recommandation H.248.1 de l'UIT-T, révisée par la Commission d'études 16 (2001-2004) de l'UIT-T, a été approuvée le 22 mai 2002 selon la procédure définie dans la Rec. UIT-T A.8.

AVANT-PROPOS

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

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

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

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© UIT 2003
Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIÈRES

		Page
1	Domaine d'application	1
2	Références	1
2.1	Références normatives	1
2.2	Références informatives	3
3	Définitions	4
4	Abréviations	5
5	Conventions	6
6	Modèle de connexion	6
6.1	Contextes	7
6.1.1	Attributs et descripteurs de contexte	8
6.1.2	Création, suppression et modification de contextes	8
6.2	Terminaisons	8
6.2.1	Dynamique de terminaison	10
6.2.2	TerminationIDs (identificateurs de terminaison)	11
6.2.3	Paquetages	11
6.2.4	Propriétés et descripteurs de terminaison	12
6.2.5	Terminaison racine	14
7	Commandes	14
7.1	Descripteurs	15
7.1.1	Spécification des paramètres	15
7.1.2	Descripteur Modem	16
7.1.3	Descripteur Mux	16
7.1.4	Descripteur Media	16
7.1.5	Descripteur TerminationState	17
7.1.6	Descripteur Stream	17
7.1.7	Descripteur LocalControl	18
7.1.8	Descripteurs Local et Remote	18
7.1.9	Descripteur d'événements	21
7.1.10	Descripteur EventBuffer (mémoire tampon d'événements)	23
7.1.11	Descripteur de signaux	23
7.1.12	Descripteur Audit	25
7.1.13	Descripteur ServiceChange (changement de service)	25
7.1.14	Descripteur DigitMap (script de numérotation)	26
7.1.15	Descripteur Statistics (de statistiques)	31
7.1.16	Descripteurs de paquetages	31

			Page
	7.1.17	Descripteurs ObservedEvents (éléments observés)	31
	7.1.18	Descripteur Topology	31
	7.1.19	Descripteur d'erreur	34
	7.2	Interface de programmation d'application pour les commandes	34
	7.2.1	Commande Add	34
	7.2.2	Commande Modify	36
	7.2.3	Commande Subtract	37
	7.2.4	Commande Move	38
	7.2.5	Commande AuditValue	39
	7.2.6	AuditCapabilities	42
	7.2.7	Commande Notify	44
	7.2.8	ServiceChange	44
	7.2.9	Manipulation et Audit d'attributs de contexte	48
	7.2.10	Syntaxe de commande générique	49
8	Transactions		49
	8.1	Paramètres communs	51
	8.1.1	Identificateurs de transaction	51
	8.1.2	Identificateurs de contexte	51
	8.2	Interface de programmation d'application pour les transactions	51
	8.2.1	Demande de transaction (TransactionRequest)	51
	8.2.2	Réponse de Transaction	52
	8.2.3	Transaction en cours (TransactionPending)	53
	8.3	Messages	54
9	Transport		54
	9.1	Ordonnancement des commandes	55
	9.2	Protection contre l'avalanche de redémarrage	55
10	Considérations relatives à la sécurité		56
	10.1	Protection des connexions de protocole	56
	10.2	Système provisoire d'en-tête AH	57
	10.3	Protection des connexions de médias	58
11	Interface de commande MG-MGC		58
	11.1	Multiples passerelles MG virtuelles	58
	11.2	Démarrage à froid	59
	11.3	Négociation de version de protocole	60
	11.4	Défaillance d'une passerelle MG	60
	11.5	Défaillance d'un contrôleur MGC	61

Page

12 Définition des paquetages 62
12.1 Directives pour la définition des paquetages 62
12.1.1 Paquetage 62
12.1.2 Propriétés 63
12.1.3 Evénements 64
12.1.4 Signaux 64
12.1.5 Statistiques 65
12.1.6 Procédures 65
12.2 Directives de définition des paramètres relatifs aux événements et signaux 65
12.3 Listes 65
12.4 Identificateurs 66
12.5 Enregistrement de paquetage 66

13 Définition des profils 66

14 Considérations relatives à l'autorité IANA 67
14.1 Paquetages 67
14.2 Codes d'erreur 67
14.3 Raisons de changement de service 68
14.4 Profils 68

Annexe A – Codage binaire du protocole 69
A.1 Codage des caractères génériques 69
A.2 Spécification de syntaxe en notation ASN.1 70
A.3 Noms des scripts de numérotation et des conduits 86

Annexe B – Codage alphanumérique du protocole 88
B.1 Codage des caractères génériques 88
B.2 Spécification en formalisme ABNF 88
B.3 Codage hexadécimal des octets 100
B.4 Séquence hexadécimale pour les octets 100

Annexe C – Etiquettes des propriétés de flux média 101
C.1 Attributs généraux des médias 102
C.2 Propriétés Mux 103
C.3 Propriétés générales de support 103
C.4 Propriétés générales du mode ATM 103
C.5 Relais de trames 106
C.6 IP 106
C.7 ATM AAL 2 106
C.8 ATM AAL 1 107

Page

C.9 Capacités du circuit support 108
C.10 Propriété AAL 5 117
C.11 Equivalents SDP 118
C.12 H.245 119

Annexe D – Transport sur protocole IP 119
D.1 Transport sur protocoles IP/UDP au moyen du verrouillage de niveau d'application 119
D.1.1 Fourniture de la capacité fonctionnelle d'une fois au plus 120
D.1.2 Identificateurs de transaction et dialogue à trois 120
D.1.3 Calcul des temporisations de retransmission 121
D.1.4 Réponses provisoires 122
D.1.5 Répétitions des demandes, réponses et acquittements 122
D.2 Utilisation de TCP 123
D.2.1 Fourniture de la capacité fonctionnelle d'une fois au plus 123
D.2.2 Identificateurs de transaction et dialogue à trois 123
D.2.3 Calcul des temporisations de retransmission 123
D.2.4 Réponses provisoires 124
D.2.5 Ordonnancement des commandes 124

Annexe E – Paquetages de base 124
E.1 Paquetage générique 124
E.1.1 Propriétés 124
E.1.2 Evénements 124
E.1.3 Signaux 126
E.1.4 Statistiques 126
E.2 Paquetage racine de base 126
E.2.1 Propriétés 126
E.2.2 Evénements 128
E.2.3 Signaux 128
E.2.4 Statistiques 128
E.2.5 Procédures 128
E.3 Paquetage de générateur de tonalités 128
E.3.1 Propriétés 129
E.3.2 Evénements 129
E.3.3 Signaux 129
E.3.4 Statistiques 129
E.3.5 Procédures 129
E.4 Paquetage de détection de tonalité 129
E.4.1 Propriétés 129
E.4.2 Evénements 130

		Page
E.4.3	Signaux	131
E.4.4	Statistiques	131
E.4.5	Procédures	131
E.5	Paquetage de générateur de tonalités DTMF de base	131
E.5.1	Propriétés	131
E.5.2	Evénements	132
E.5.3	Signaux	132
E.5.4	Statistiques	132
E.5.5	Procédures	132
E.6	Paquetage de détection de tonalités DTMF	133
E.6.1	Propriétés	133
E.6.2	Evénements	133
E.6.3	Signaux	134
E.6.4	Statistiques	134
E.6.5	Procédures	134
E.7	Paquetage de générateur de tonalités de progression d'appel	134
E.7.1	Propriétés	134
E.7.2	Evénements	134
E.7.3	Signaux	135
E.7.4	Statistiques	135
E.7.5	Procédures	135
E.8	Paquetage de détection de tonalités de progression d'appel	135
E.8.1	Propriétés	136
E.8.2	Evénements	136
E.8.3	Signaux	136
E.8.4	Statistiques	136
E.8.5	Procédures	136
E.9	Paquetage de supervision de ligne analogique	136
E.9.1	Propriétés	136
E.9.2	Evénements	136
E.9.3	Signaux	138
E.9.4	Statistiques	138
E.9.5	Procédures	139
E.9.6	Code d'erreur	139
E.10	Paquetage de continuité de base	139
E.10.1	Propriétés	139
E.10.2	Evénements	139
E.10.3	Signaux	139
E.10.4	Statistiques	140
E.10.5	Procédures	140

Page

E.11 Paquetage de réseau 140
E.11.1 Propriétés 141
E.11.2 Evénements 141
E.11.3 Signaux 142
E.11.4 Statistiques 142
E.11.5 Procédures 142
E.12 Paquetage du protocole de transport en temps réel (RTP) 142
E.12.1 Propriétés 142
E.12.2 Evénements 142
E.12.3 Signaux 143
E.12.4 Statistiques 143
E.12.5 Procédures 143
E.13 Paquetage de circuit TDM 144
E.13.1 Propriétés 144
E.13.2 Evénements 144
E.13.3 Signaux 144
E.13.4 Statistiques 144
E.13.5 Procédures 144

Appendice I – Exemples de flux de communication 145
I.1 Communication de passerelle résidentielle à passerelle résidentielle 145
I.1.1 Programmation du comportement de repos de terminaisons de ligne analogique passant par une passerelle résidentielle 145
I.1.2 Acquisition des chiffres de l'expéditeur et de la terminaison d'origine 146

Recommandation UIT-T H.248.1

Protocole de commande de passerelle: version 2

1 Domaine d'application

La présente Recommandation définit les protocoles utilisés entre les éléments de décomposition physique d'une passerelle multimédia. Il n'y a pas de différences fonctionnelles, en termes de système, entre une passerelle décomposée (comportant des sous-éléments théoriquement répartis sur plusieurs unités physiques) et une passerelle monolithique comme celle qui est décrite dans la Rec. UIT-T H.246. La présente Recommandation ne définit pas la façon dont fonctionnent les passerelles, les ponts de conférence ou les répondeurs vocaux interactifs (IVR, *interactive voice response*) mais elle crée un cadre général qui convient à ces applications.

Les interfaces avec les réseaux en mode paquet peuvent être de type IP, ATM, etc. Ces interfaces prendront en charge divers systèmes de signalisation de réseau à commutation de circuits (RCC) dont la signalisation par fréquences vocales, le RNIS, le sous-système utilisateur du RNIS, la signalisation à l'interface Q et le système GSM. Les variantes nationales de ces systèmes de signalisation seront éventuellement prises en charge.

Les produits pour lesquels il est indiqué qu'ils sont conformes à la version 1 de la Rec. UIT-T H.248.1 doivent être conformes à toutes les spécifications obligatoires de la Rec. UIT-T, approuvée initialement en juin 2000, et rééditée en mars 2002.

Les produits pour lesquels il est indiqué qu'ils sont conformes à la présente Recommandation doivent être conformes à toutes les spécifications obligatoires de la Rec. UIT-T, approuvée en mai 2002.

Les produits doivent porter la mention de la version du protocole employé. La valeur du paramètre ServiceChangeVersion sera "1" pour la version de la Rec. UIT-T H.248.1 de mars 2002 et "2" pour celle de la Rec. UIT-T H.248.1 de mai 2002.

2 Références

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

2.1 Références normatives

– Recommandation UIT-T H.225.0 (2000), *Protocoles de signalisation d'appel et paquétisation des flux monomédias dans les systèmes de communication multimédias en mode paquet.*

– Recommandation UIT-T H.235 (2000), *Sécurité et cryptage des terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245).*

– Recommandation UIT-T H.245 (2001), *Protocole de commande pour communications multimédias.*

– Recommandation UIT-T H.246 (1998), *Interfonctionnement des terminaux multimédias de la série H avec d'autres terminaux multimédias de la série H et des terminaux vocaux ou en bande vocale sur le RTGC et le RNIS.*

– Recommandation UIT-T H.248.4 (2000), *Protocole de commande de passerelle: transport par le protocole de transmission de commande de flux (SCTP).*

– Recommandation UIT-T H.248.5 (2000), *Protocole de commande de passerelle: transport sur réseau ATM.*

– Recommandation UIT-T H.248.8 (2002), *Protocole de commande de passerelle: description des codes d'erreur et des causes de changement de service.*

– Recommandation UIT-T H.323 (2000), *Systèmes de communication multimédia en mode paquet*.

– Recommandation UIT-T I.363.1 (1996), *Spécification de la couche d'adaptation ATM du RNIS-LB: AAL de type 1*.

– Recommandation UIT-T I.363.2 (2000), *Spécification de la couche d'adaptation ATM du RNIS-LB: AAL de type 2*.

– Recommandation UIT-T I.363.5 (1996), *Spécification de la couche d'adaptation ATM du RNIS LB: AAL de type 5*.

– Recommandation UIT-T I.366.1 (1998), *Sous-couche de convergence propre au service de segmentation et de réassemblage pour la couche d'adaptation ATM de type 2.*

– Recommandation UIT-T I.366.2 (2000), *Sous-couche de convergence propre au service de la couche AAL de type 2 pour les services à bande étroite*.

– Recommandation UIT-T I.371 (2000), *Gestion du trafic et des encombrements dans le RNIS-LB.*

– Recommandation UIT-T Q.763 (1999), *Système de signalisation n° 7 – Formats et codes du sous-système utilisateur du RNIS.*

– Recommandation UIT-T Q.765.5 (2000), *Système de signalisation n° 7 – Mécanisme de transport d'application: commande d'appel indépendante du support.*

– Recommandation UIT-T Q.931 (1998), *Spécification de la couche 3 de l'interface utilisateur-réseau RNIS pour la commande de l'appel de base.*

– Recommandation UIT-T Q.2630.1 (1999), *Protocole de signalisation de couche AAL de type 2 (ensemble de capacités 1).*

– Recommandation UIT-T Q.2931 (1995), *Système de signalisation d'abonné numérique n° 2 – Spécification de la couche 3 de l'interface utilisateur-réseau pour la commande de connexion/appel de base.*

– Recommandation UIT-T Q.2941.1 (1997), *Système de signalisation d'abonné numérique n° 2 – Transport des identificateurs génériques.*

– Recommandation UIT-T Q.2961.1 (1995), *Système de signalisation d'abonné numérique n° 2 – Paramètres de trafic supplémentaires: Capacités de signalisation supplémentaires pour la prise en charge des paramètres de trafic relatifs à l'option d'étiquetage et au jeu de paramètres de débit cellulaire permanent acceptable.*

– Recommandation UIT-T Q.2961.2 (1997), *Système de signalisation d'abonné numérique n° 2 – Paramètres de trafic supplémentaires: prise en charge de la capacité de transfert ATM dans l'élément d'information de capacité de support à large bande.*

– Recommandation UIT-T Q.2965.1 (1999), *Système de signalisation d'abonné numérique n° 2 – Prise en charge des classes de qualité de service*.

– Recommandation UIT-T Q.2965.2 (1999), *Système de signalisation d'abonné numérique n° 2 – Signalisation des paramètres de qualité de service individuelle*.

– Recommandation UIT-T V.76 (1996), *Multiplexeur générique utilisant les procédures basées LAPM de la Recommandation V.42*.

– Recommandation UIT-T X.213 (2001) | ISO/CEI 8348:2002, *Technologies de l'information – Interconnexion des systèmes ouverts – Définition du service réseau* plus Amendement 1 (1997), *Adjonction de l'identificateur de format d'adresse du protocole Internet*.

– Recommandation UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.

– Recommandation UIT-T X.690 (2002) | ISO/CEI 8825-1:2002, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*

– ATM Forum (1996), *ATM User-Network Interface (UNI) Signalling Specification – Version 4.0.*

– IETF RFC 1006 (1987), *ISO Transport Service on top of the TCP, Version 3* (Service de transport ISO au-dessus du protocole TCP, version 3).

– IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF* (Développement du formalisme BNF pour les spécifications en notation syntaxique: le formalisme ABNF).

– IETF RFC 2327 (1998), *SDP: Session Description Protocol* (Protocole de description de session: SDP).

– IETF RFC 2402 (1998), *IP Authentication Header* (En-tête d'authentification IP).

– IETF RFC 2406 (1998), *IP Encapsulating Security Payload (ESP)* [Charge utile de sécurité par encapsulation IP (ESP)].

2.2 Références informatives

– Recommandation UIT-T E.180/Q.35 (1998), *Caractéristiques techniques des tonalités du service téléphonique*.

– Recommandation UIT-T G.711 (1988), *Modulation par impulsions et codage (MIC) des fréquences vocales*.

– Recommandation UIT-T H.221 (1999), *Structure de trame pour un canal d'un débit de 64 à 1920 kbit/s pour les téléservices audiovisuels*.

– Recommandation UIT-T H.223 (2001), *Protocole de multiplexage pour communications multimédias à faible débit*.

– Recommandation UIT-T H.226 (1998), *Protocole d'agrégation de canaux pour l'exploitation en mode multiliaison sur les réseaux à commutation de circuits.*

– Recommandation UIT-T Q.724 (1998), *Procédures de signalisation du sous-système utilisateur de téléphonie.*

– Recommandation UIT-T Q.764 (1999), *Système de signalisation n° 7 – Procédures de signalisation du sous-système utilisateur du RNIS*.

– Recommandation UIT-T Q.1904.2 (2001), *Protocole de commande d'appel indépendante du support (ensemble de capacités 2): procédures de commande de base.*

– IETF RFC 768 (1980), *User Datagram Protocol.*

– IETF RFC 791 (1981), *Internet protocol.*

– IETF RFC 793 (1981), *Transmission control protocol.*

– IETF RFC 1661 (1994), *The Point-to-Point Protocol (PPP)* (le protocole point à point).

– IETF RFC 1889 (1996), *RTP: A Transport Protocol for Real-Time Applications* (RTP: un protocole de transport pour applications en temps réel).

– IETF RFC 1890 (1996), *RTP Profile for Audio and Video Conferences with Minimal Control* (Profil RTP pour conférences audio et vidéo avec commandes minimales).

– IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol* (Architecture de sécurité pour le protocole Internet).

– IETF RFC 2543 (1999), *SIP: Session Initiation Protocol* (SIP: protocole d'ouverture de session).

– IETF RFC 2460 (1998), *Internet Protocol, Version 6 (IPv6) Specification.*

– IETF RFC 2805 (2000), *Media Gateway Control Protocol Architecture and Requirements* (Architectures et prescriptions pour le protocole de commande de passerelle de média).

3 Définitions

La présente Recommandation définit les termes suivants:

3.1 passerelle d'accès: type de passerelle offrant une interface de type utilisateur-réseau (UNI), comme pour le RNIS.

3.2 descripteur: élément syntaxique du protocole qui regroupe des propriétés associées. Par exemple, le contrôleur de passerelle média (MGC) peut régler les propriétés d'un flux média en insérant le descripteur approprié dans une commande.

3.3 passerelle média (MG, *media gateway*): passerelle qui convertit le format du média fourni par un type de réseau de façon qu'il soit accepté par un autre type de réseau. Par exemple, une passerelle média peut fermer des voies supports issues d'un réseau à commutation de circuits (des signaux DS0 par exemple) ainsi que des flux média issus d'un réseau en mode paquet (par exemple des flux RTP dans un réseau IP). Cette passerelle peut traiter des signaux audio, vidéo et T.120 isolés ou combinés d'une manière quelconque. Elle pourra également assurer des conversions de média en duplex, restituer des messages audio/vidéo, remplir d'autres fonctions de répondeur IVR ou assurer des conférences multimédias.

3.4 contrôleur de passerelle média (MGC, *media gateway controller*): entité qui commande les parties de l'état d'appel qui correspondent à la commande de connexion pour les voies média d'une passerelle MG.

3.5 pont de conférence; **unité de commande multipoint (MCU, *multipoint control unit*)**: entité qui commande l'établissement et la coordination d'une conférence entre plusieurs utilisateurs comportant normalement le traitement audio, vidéo et données.

3.6 passerelle résidentielle: passerelle qui assure l'interfonctionnement d'une ligne analogique avec un réseau en mode paquet. Normalement, une passerelle résidentielle contient une ou deux lignes analogiques et est située dans les locaux d'abonné.

3.7 passerelle de signalisation de réseau RCC en mode service par service: fonction contenant l'interface de signalisation de réseau RCC à laquelle aboutissent des canaux sémaphores SS7, RNIS ou d'autres canaux dans lesquels la voie de commande d'appel et les voies supports sont copositionnées dans le même arc physique.

3.8 passerelle de signalisation de réseau RCC en mode autre que service par service: fonction contenant l'interface de signalisation de réseau RCC à laquelle aboutissent des canaux sémaphores SS7 ou d'autres canaux dans lesquels les voies de commande d'appel sont séparées des voies supports.

3.9 flux: média bidirectionnel ou débit de commande reçu/envoyé par une passerelle média dans le cadre d'une communication ou d'une conférence.

3.10 jonction: canal de communication entre deux systèmes de commutation comme un signal DS0 sur une ligne à débit T1 ou E1.

3.11 passerelle de jonction: passerelle entre un réseau RCC et un réseau en mode paquet à laquelle aboutissent normalement un grand nombre de circuits numériques.

4 Abréviations

La présente Recommandation utilise les abréviations suivantes:

ALF verrouillage au niveau application (*application level framing*)

ATM mode de transfert asynchrone (*asynchronous transfer mode*)

CAS signalisation en mode voie par voie (*channel associated signalling*)

DTMF multifréquence bitonalité (*dual tone multi-frequency*)

FAS signalisation en mode service par service (*facility associated signalling*)

GSM système mondial de communications mobiles (*global system for mobile communications*)

GW passerelle (*gateway*)

IANA Autorité chargée de l'assignation des numéros Internet (*Internet assigned numbers authority*) remplacée par ICANN, Internet Corporation for assigned names and numbers

IP protocole Internet (*Internet protocol*)

ISUP sous-système utilisateur du RNIS (*ISDN user part*)

IVR réponse vocale interactive (*interactive voice response*)

MG passerelle média (*media gateway*)

MGC contrôleur de passerelle média (*media gateway controller*)

NFAS signalisation autre qu'en mode service par service (*non-facility associated signalling*)

PRI interface à débit primaire (*primary rate interface*)

QS qualité de service

RCC réseau à commutation de circuits

RTP protocole de transport en temps réel (*real-time transport protocol*)

RTPC réseau téléphonique public commuté

SG passerelle de signalisation (*signalling gateway*)

SS7 système de signalisation n° 7

5 Conventions

Dans la présente Recommandation, la forme verbale "DOIT" correspond à une exigence obligatoire, tandis que la forme "DEVRAIT" ou "IL CONVIENT" correspond à la suggestion d'un élément de service ou d'une procédure, mais à titre facultatif. La forme "PEUT" correspond à une action possible sans qu'une préférence soit exprimée.

6 Modèle de connexion

Le modèle de connexion pour le protocole décrit les entités logiques – ou objets contenus dans la passerelle média, que le contrôleur de passerelle média peut commander. Les principales abstractions utilisées dans ce modèle de connexion sont les terminaisons et les contextes.

Une *terminaison* envoie et/ou collecte un ou plusieurs flux. Dans une conférence multimédia, une terminaison peut être de type multimédia et peut envoyer ou collecter de multiples flux médias. Les paramètres des flux médias, ainsi que les paramètres de support sont encapsulés dans la terminaison.

Un *contexte* est un paquetage de terminaisons associées. Il existe un type spécial de contexte, le contexte *néant (null)*, qui contient toutes les terminaisons non associées à une autre terminaison. Par exemple, dans une passerelle d'accès décomposée, toutes les lignes au repos sont représentées par des terminaisons du contexte néant.

Une description graphique de ces concepts est donnée dans le schéma de la Figure 1, qui donne plusieurs exemples et qui ne vise pas à illustrer tous les cas. L'astérisque encadré de chaque contexte représente l'association logique des terminaisons appartenant au contexte.

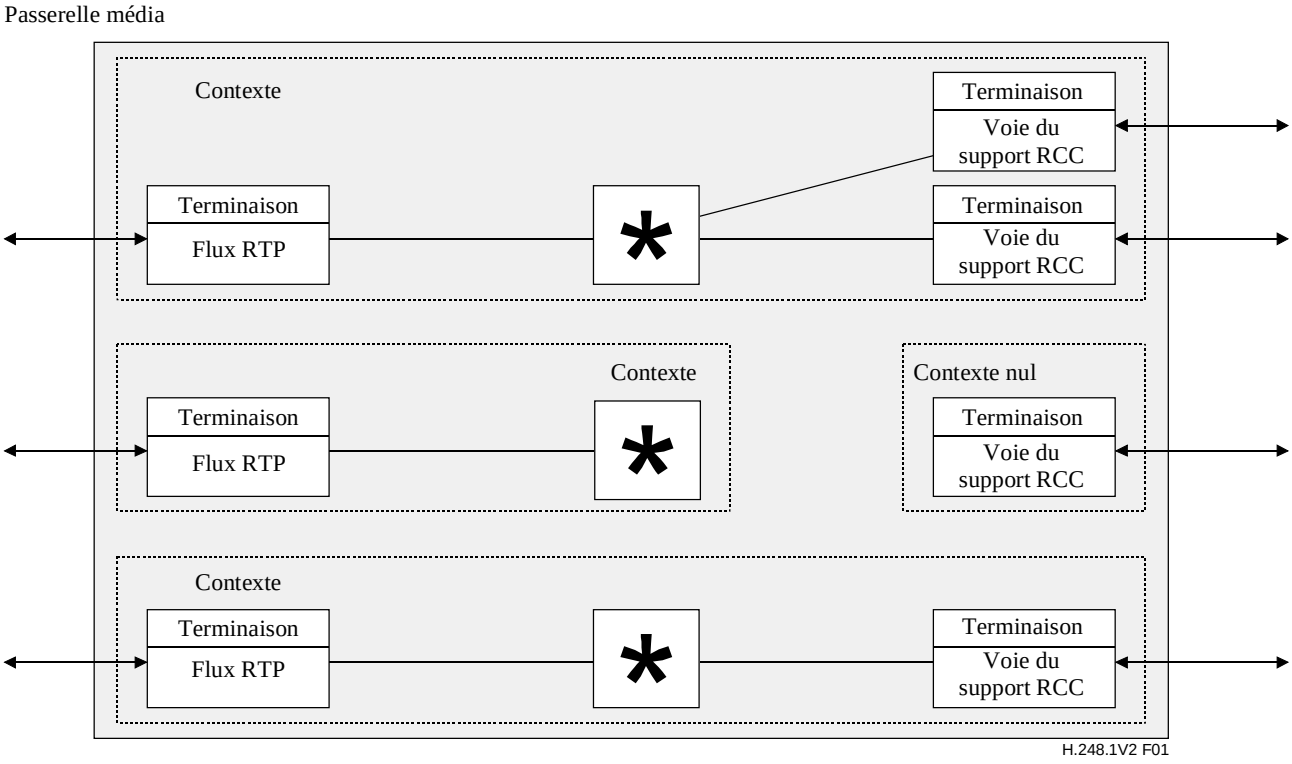


Figure 1/H.248.1 – Exemple de modèle de connexion H.248.1

L'exemple dans la Figure 2 montre un mode de réalisation d'un scénario de signal d'appel dans une passerelle d'accès décomposée, illustrant la relocalisation d'une terminaison entre plusieurs contextes. Les terminaisons T1 et T2 appartiennent au contexte C1 dans une communication audio

bilatérale. Une deuxième communication audio est en instance pour la terminaison T1 en provenance de la terminaison T3, qui est seule dans le contexte C2. T1 accepte l'appel de T3 et met T2 en instance. Cette action se traduit par le passage de T1 dans le contexte C2, comme représenté dans la Figure 3.

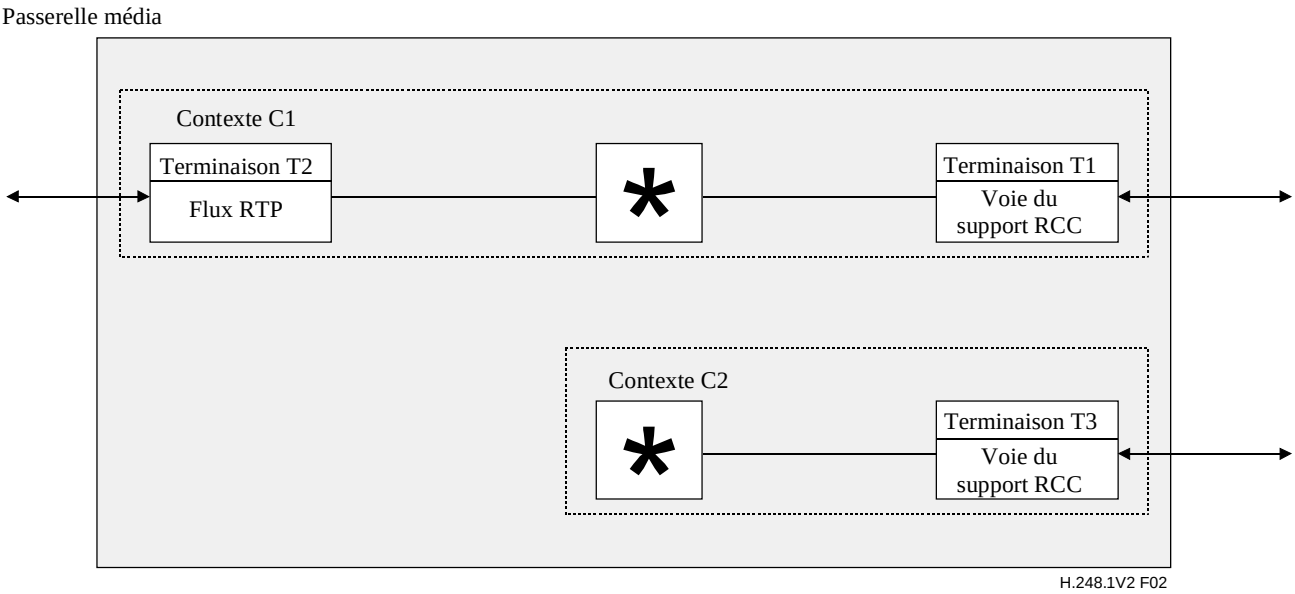


Figure 2/H.248.1 – Exemple de scénario de signal d'appel/sonnerie appliqué à la terminaison T1

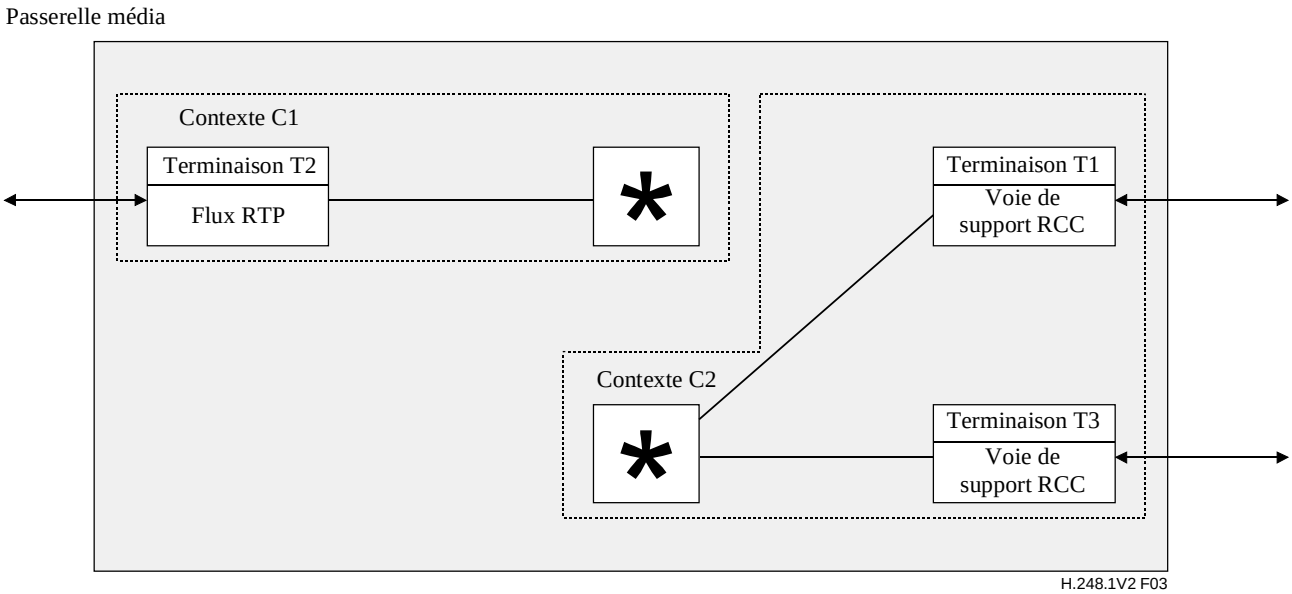


Figure 3/H.248.1 – Exemple de scénario de signal d'appel/réponse appliqué à la terminaison T1

6.1 Contextes

Un contexte est une association entre un certain nombre de terminaisons. Le contexte décrit la topologie (qui entend/voit qui) et les paramètres de mélange et/ou de commutation de média si plusieurs terminaisons font partie de l'association.

Il existe un contexte spécial appelé contexte *néant* qui contient les terminaisons qui ne sont associées à aucune autre terminaison. Dans le contexte néant, les terminaisons peuvent avoir leurs paramètres examinés ou modifiés et des événements peuvent être détectés à leur sujet.

En général, une commande d'addition (Add) permet d'ajouter des terminaisons à un contexte. Si le contrôleur MGC ne spécifie pas un contexte existant auquel la terminaison doit être ajoutée, la passerelle MG crée un nouveau contexte. Une terminaison peut être retirée d'un contexte par une commande de soustraction (Subtract) et une terminaison peut être déplacée d'un contexte à un autre par une commande de déplacement (Move). Une terminaison ne DOIT exister que dans un seul contexte à la fois.

Le nombre maximal de terminaisons contenues dans un contexte est une propriété de passerelle MG. Les passerelles MG qui n'offrent qu'une connexité point à point peuvent admettre au plus deux terminaisons par contexte. Les passerelles MG qui prennent en charge les conférences multipoint peuvent admettre au moins trois terminaisons par contexte.

6.1.1 Attributs et descripteurs de contexte

Les attributs des contextes sont les suivants:

- ContextID;
- la topologie (qui entend/voit qui);

  la topologie d'un contexte décrit le flux média entre les terminaisons contenues dans un contexte. En revanche, le mode d'une terminaison (émission/réception/…) décrit le flux média à l'entrée/la sortie de la passerelle média;
- la priorité accordée à un contexte afin de fournir à la passerelle MG des informations sur un certain traitement de préséance pour un contexte. Le contrôleur MGC peut également utiliser la priorité pour commander de manière autonome et souple la préséance de trafic dans la passerelle MG lors de certaines situations (comme un redémarrage) où un grand nombre de contextes doivent être traités simultanément. La priorité 0 est la plus basse priorité tandis que la priorité 15 est la plus haute priorité;
- un indicateur d'appel d'urgence fourni également pour permettre un traitement préférentiel dans la passerelle MG.

6.1.2 Création, suppression et modification de contextes

Le protocole peut être utilisé pour (implicitement) créer des contextes et modifier les valeurs paramétriques des contextes existants. Ce protocole contient des commandes permettant d'ajouter des terminaisons à des contextes, de retirer des terminaisons de contextes et de déplacer des terminaisons entre contextes. Lorsque la dernière terminaison restante est soustraite ou déplacée, son contexte est implicitement supprimé.

6.2 Terminaisons

Une terminaison est une entité logique d'une passerelle MG qui envoie et collecte des flux de média et de commande. Une terminaison est décrite par un certain nombre de propriétés de caractérisation qui sont groupées dans un paquetage de descripteurs utilisés dans les commandes. Les terminaisons ont des identificateurs uniques (TerminationID) qui sont attribués par la passerelle MG au moment de la création de ces terminaisons.

Les terminaisons qui représentent des entités physiques ont une existence semi-permanente. Par exemple, une terminaison représentant une voie TDM peut exister aussi longtemps qu'elle est fournie dans la passerelle. Les terminaisons représentant des flux informationnels éphémères, comme les flux de protocole RTP, n'existeront généralement que pendant leur durée d'utilisation.

Les terminaisons éphémères sont créées au moyen d'une commande Add. Elles sont détruites au moyen d'une commande Subtract. En revanche, lorsqu'une terminaison physique est ajoutée à un contexte ou en est soustraite, elle est respectivement extraite du contexte néant ou placée dans celui-ci.

Les terminaisons peuvent subir l'application de signaux (se reporter au § 7.1.11). Les terminaisons peuvent être programmées de façon à détecter des événements, dont l'apparition peut déclencher l'envoi de messages de notification vers le contrôleur MGC ou déclencher une action de la passerelle MG. Des statistiques peuvent être cumulées au sujet d'une terminaison. Ces statistiques sont signalées au contrôleur MGC sur demande (au moyen d'une commande AuditValue, se reporter au § 7.2.5) et lorsque la terminaison est soustraite d'un contexte.

Les passerelles MG peuvent traiter des flux médias multiplexés. Par exemple, la Rec. UIT-T H.221 décrit une structure de trame pour le multiplexage de plusieurs flux médias dans un certain nombre de voies numériques à 64 kbit/s. Un tel cas est pris en charge comme suit dans le modèle de connexion. Il existe une "terminaison support" physique ou éphémère pour chaque canal support qui achemine une partie des flux multiplexés. Les terminaisons supports qui envoient/collectent ces voies numériques sont connectées à une terminaison distincte appelée "terminaison de multiplexage". La terminaison de multiplexage est une terminaison éphémère représentant une session orientée trame. Le descripteur de multiplex MultiplexDescriptor de cette terminaison décrit le multiplex utilisé (par exemple H.221 pour une session H.320) et indique l'ordre dans lequel les voies numériques contenues sont assemblées en une trame.

Les terminaisons de multiplexage peuvent être placées en cascade (par exemple un multiplex H.226 de voies numériques alimentant un multiplex H.223 qui prend en charge une session H.324).

Les flux médias individuels acheminés au cours de la session sont décrits par les descripteurs de flux StreamDescriptors au niveau de la terminaison de multiplexage. Ces flux médias peuvent être associés à des flux envoyés/collectés par des terminaisons du contexte autres que les terminaisons support prenant en charge la terminaison de multiplexage. Toutes les terminaisons supports prennent en charge un seul flux de données. Ces flux de données n'apparaissent pas explicitement comme flux au niveau de la terminaison de multiplexage et ils sont cachés du reste du contexte.

Les Figures 4, 5 et 6 représentent des applications typiques de la terminaison de multiplexage et du scripteur de multiplex.

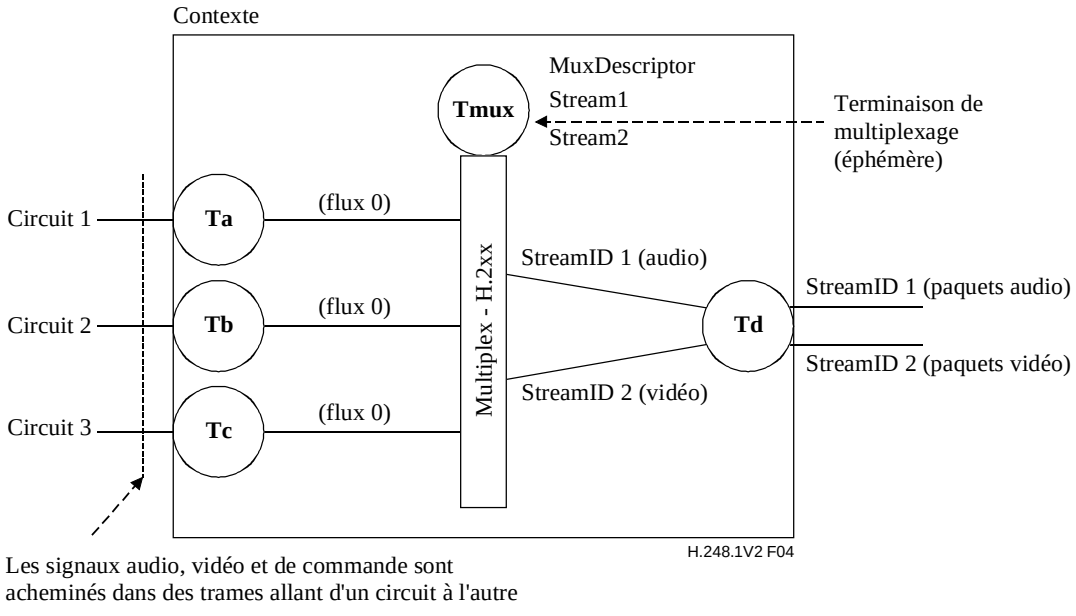


Figure 4/H.248.1 – Scénario de terminaison multiplexée – De circuits à paquets

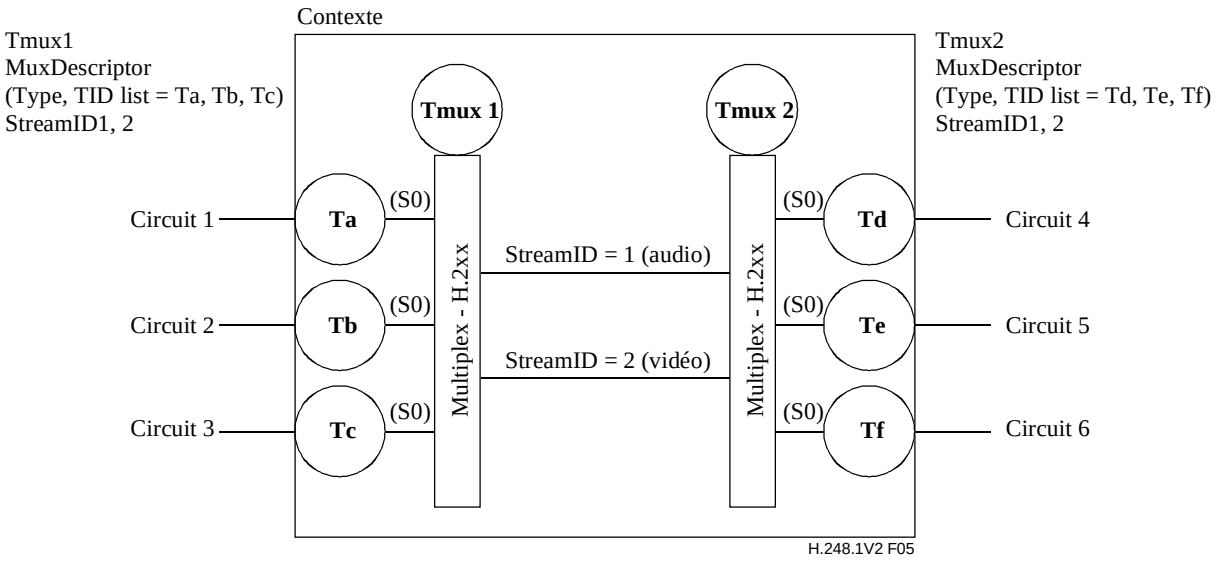


Figure 5/H.248.1 – Scénario de terminaison multiplexée – De circuits à circuits

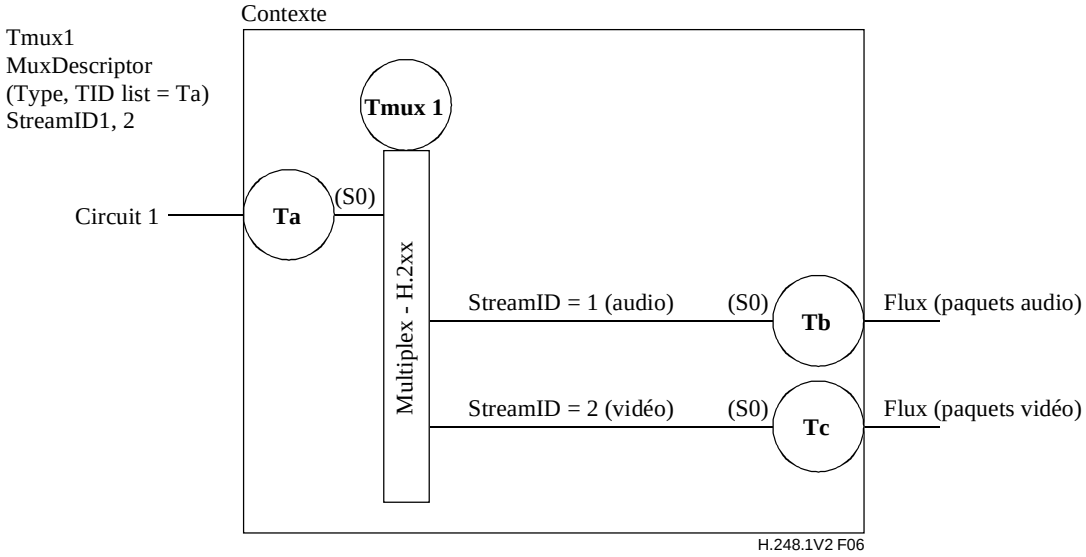


Figure 6/H.248.1 – Scénario de terminaison multiplexée – De terminaison simple à terminaisons multiples

Contrairement aux terminaisons de multiplexage décrites dans le paragraphe précédent, les terminaisons support multiplexées correspondant aux supports multiplexés tels que les supports en mode ATM AAL de type 2, n'acheminent pas de flux média. Elles ne servent qu'à modéliser la création et la destruction du support proprement dit. Lorsqu'un circuit multiplexé doit être créé, une terminaison éphémère est créée dans un contexte établi à cet effet. Lorsque la terminaison est soustraite, le circuit support multiplexé est détruit.

6.2.1 Dynamique de terminaison

Le protocole peut être utilisé pour créer de nouvelles terminaisons et pour modifier les valeurs des propriétés de terminaisons existantes. Ces modifications comprennent la possibilité d'ajouter ou de supprimer des événements et/ou des signaux. Les propriétés des terminaisons ainsi que les événements et les signaux sont décrits dans les paragraphes qui suivent. Un contrôleur MGC ne

peut libérer/modifier que les terminaisons (et les ressources qu'elles représentent) qui ont été préalablement saisies au moyen, par exemple, de la commande Add.

6.2.2 TerminationIDs (identificateurs de terminaison)

Les terminaisons sont désignées par un identificateur de terminaison qui est une séquence arbitraire, choisie par la passerelle MG.

Les identificateurs de terminaisons physiques sont fournis dans la passerelle MG. Ils peuvent être choisis de façon à posséder une structure. Par exemple, un identificateur de terminaison peut se composer d'un faisceau de circuits et d'une jonction locale dans ce faisceau.

Un mécanisme de remplacement par des caractères génériques, utilisant deux types de caractère générique, peut être utilisé avec les identificateurs de terminaison. Ces deux caractères sont ALL et CHOOSE. Le premier sert à désigner simultanément plusieurs terminaisons tandis que le second sert à indiquer à une passerelle MG qu'elle doit sélectionner une terminaison correspondant à l'identificateur de terminaison partiellement spécifié. Cela permet à un contrôleur MGC de demander, par exemple, à une passerelle MG de choisir un circuit dans un faisceau de circuits.

Si le caractère ALL est utilisé dans l'identificateur de terminaison d'une commande, l'effet est identique à une répétition de la commande avec chacun des identificateurs de terminaison réels qui correspondent. L'emploi de ALL n'inclut pas la terminaison ROOT. Etant donné que chacune de ces commandes peut générer une réponse, la taille de la réponse complète peut être importante. Si des réponses individuelles ne sont pas requises, une réponse générique peut être demandée. Dans ce cas, une seule réponse est générée et elle contient l'UNION de toutes les réponses individuelles qui auraient été autrement générées, les valeurs répétées étant supprimées. Par exemple, étant donné une terminaison Ta dont les propriétés seraient p1 = a, p2 = b et une terminaison Tb dont les propriétés seraient p2 = c, p3 = d, une réponse UNION contiendrait un identificateur de terminaison remplacé par un caractère générique et la séquence de propriétés p1 = a, p2 = b,c et p3 = d. La réponse générique peut être particulièrement utile dans les commandes d'Audit.

Le codage du mécanisme de remplacement par des caractères génériques est détaillé dans les Annexes A et B.

6.2.3 Paquetages

Différents types de passerelle peuvent implémenter des terminaisons possédant des caractéristiques très différentes. Le protocole tient compte des variantes de terminaison en permettant que leurs implémentations par des passerelles MG aient des propriétés, des événements, des signaux et des statistiques de type facultatif.

Afin de réaliser l'interopérabilité MG/MGC, de telles options sont groupées en paquetages. Une terminaison réalise en règle générale un surensemble de tels paquetages. Pour plus d'informations sur la définition des paquetages, se reporter au paragraphe 12. Un contrôleur MGC peut auditer une terminaison afin de déterminer les paquetages qu'elle réalise.

Les propriétés, événements, signaux et statistiques définis dans les paquetages, ainsi que des paramètres les concernant, sont référencés par des identificateurs (Id). Les identificateurs ont des portées. Pour chaque paquetage, les identificateurs PropertyId, EventId, SignalId, StatisticsId et ParameterId ont des espaces nominatifs uniques et le même identificateur peut être utilisé dans chacun d'eux. Deux identificateurs PropertyId présents dans des paquetages différents peuvent également avoir le même identificateur, etc.

Afin de prendre en charge un paquetage particulier, la passerelle MG doit prendre en charge l'ensemble des propriétés, signaux, événements et statistiques définis dans un paquetage. Elle doit aussi prendre en charge un sous-ensemble de valeurs énumérées dans un paquetage pour une propriété ou un paramètre particulier.

Lorsque les paquetages sont étendus, il peut être renvoyé aux propriétés, événements, signaux et statistiques définis dans le paquetage de base au moyen soit du nom de paquetage étendu, soit du nom de paquetage de base. Par exemple, si le paquetage A définit un événement e1 et que le paquetage B étend le paquetage A, alors B/e1 est un événement pour une terminaison implémentant le paquetage B. Par définition, la passerelle MG DOIT aussi implémenter le paquetage de base, mais elle peut choisir de publier ou non le paquetage de base en tant qu'interface admise. Si elle publie le paquetage A, ce paquetage tout comme le paquetage B sera indiqué dans le descripteur de paquetage dans la valeur AuditValue, et l'événement A/e1 serait disponible sur une terminaison. Si la passerelle MG ne publie pas le paquetage A, seul l'événement B/e1 serait disponible. Si la publication se fait au moyen de la valeur AuditValue, les événements A/e1 et B/e1 sont le même événement.

Pour améliorer l'interfonctionnement et la compatibilité vers l'arrière, une passerelle MG PEUT publier tous les paquetages pris en charge par ses terminaisons, y compris les paquetages de base à partir desquels les paquetages étendus sont définis. Ceci peut se faire sauf dans les cas où les paquetages de base sont expressément "Conçus uniquement pour être étendus".

6.2.4 Propriétés et descripteurs de terminaison

Les terminaisons possèdent des propriétés qui ont des identificateurs PropertyID uniques. La plupart des propriétés ont des valeurs par défaut, qui sont définies de manière explicite dans la présente spécification de protocole ou dans un paquetage (voir le paragraphe 12) ou définies par la fourniture. Si elles ne sont pas fournies autrement, les propriétés dans tous les descripteurs, à l'exception de TerminationState et LocalControl, possèdent par défaut la valeur vide/"pas de valeur" lorsqu'une terminaison est créée ou retourne au contexte néant. Les contenus par défaut de ces deux exceptions sont décrits aux § 7.1.5 et 7.1.7.

La fourniture d'une valeur de propriété dans la passerelle MG va permettre à toute valeur par défaut d'être supplantée, que celle-ci soit fournie dans la spécification de protocole ou dans un paquetage. Donc, s'il est essentiel pour le contrôleur MGC de contrôler entièrement les valeurs des propriétés d'une terminaison, il devrait fournir des valeurs explicites lorsqu'il effectue l'opération ADD d'une terminaison à un contexte. Alternativement, pour une terminaison physique, il peut déterminer les valeurs des propriétés fournies en examinant la terminaison tandis que celle-ci est dans le contexte NULL.

Il existe un certain nombre de propriétés communes aux terminaisons et des propriétés spécifiques aux flux médias. Ces propriétés communes sont également appelées propriétés d'état de terminaison. Pour chaque flux média, il existe des propriétés locales et des propriétés propres aux flux reçus et émis.

Les propriétés non incluses dans le protocole de base sont définies dans des paquetages. Elles sont désignées par un nom formé du nom de paquetage (PackageName) et de l'identificateur de propriété (PropertyId). La plupart des propriétés ont des valeurs par défaut, qui figurent dans la description de paquetage. Les propriétés peuvent être de type à lecture seulement ou à lecture-écriture. Les valeurs possibles d'une propriété peuvent être analysées, ainsi que leurs valeurs actuelles. Dans le cas des propriétés qui sont accessibles en lecture/écriture, le contrôleur MGC peut en régler la valeur. On peut déclarer "Globale" une propriété qui a une seule valeur partagée par toutes les terminaisons réalisant le paquetage. Par commodité, les propriétés associées sont groupées en descripteurs.

Lorsqu'une terminaison est ajoutée à un contexte, la valeur de ses propriétés en lecture/écriture peut être réglée par inclusion dans la commande Add des descripteurs appropriés sous forme de paramètres. De même, une propriété de terminaison contenue dans un contexte peut subir une modification de valeur par la commande Modify. Les valeurs des propriétés peuvent être également modifiées lorsqu'une terminaison est déplacée d'un contexte à un autre à la suite d'une commande Move. Dans certains cas, des descripteurs sont fournis en tant que données de sortie d'une commande.

En général, lorsqu'un descripteur ne figure pas du tout dans l'une des commandes susmentionnées, les propriétés dans ce descripteur conservent leurs valeurs antérieures pour la ou les terminaisons sur lesquelles agit la commande. D'autre part, si certaines propriétés de lecture/écriture ne figurent pas dans un descripteur dans une commande (par, le descripteur n'est que partiellement spécifié), ces propriétés reprendront leurs valeurs par défaut pour la ou les terminaisons sur lesquelles agit la commande, à moins que le paquetage n'indique un autre comportement. Pour plus de détails, se reporter au § 7.1 qui traite des différents descripteurs.

Le tableau ci-dessous énumère tous les descripteurs possibles ainsi que leur usage. Tous les descripteurs ne sont pas autorisés en tant que paramètres d'entrée ou de sortie d'une commande donnée.

Nom du descripteur	**Description**
Modem	Désigne, le cas échéant, le type et les propriétés d'un modem (Note).
Mux	Décrit le type de multiplex pour des terminaisons multimédias (par exemple de type H.221, H.223, H.225.0) et pour des terminaisons formant le multiplex d'entrée.
Media	Liste de spécifications de flux média (se reporter au § 7.1.4).
TerminationState	Propriétés d'une terminaison qui peuvent être définies dans des paquetages et qui ne sont pas spécifiques d'un flux.
Stream	Liste de descripteurs de commande distante ou locale (de type remote/local//localControl) pour un flux isolé.
Local	Contient des propriétés qui spécifient les flux médias que la passerelle MG reçoit de l'entité distante.
Remote	Contient des propriétés qui spécifient les flux média que la passerelle MG envoie à l'entité distante.
LocalControl	Contient des propriétés qui peuvent être définies dans des paquetages et qui sont utiles entre la passerelle MG et le contrôleur MGC.
Events	Décrit des événements qui doivent être détectés par la passerelle MG et la suite qui doit être donnée lors d'une telle détection.
EventBuffer	Décrit des événements qui doivent être détectés par la passerelle MG lorsqu'une mise en mémoire tampon d'événement est active.
Signals	Décrit des signaux (se reporter au § 7.1.11) appliqués aux terminaisons.
Audit	Désigne, dans des commandes de type Audit, les informations à rechercher.
Packages	Renvoie, dans une commande AuditValue, une liste de paquetages réalisés par une terminaison.
DigitMap	Définit des structures avec lesquelles des séquences d'un ensemble spécifié d'événements doivent être mises en correspondance de telle sorte qu'elles puissent être rapportées en tant que groupe et non individuellement
ServiceChange	Indique, dans la commande ServiceChange, la nature, la raison, etc., d'un éventuel changement de service.
ObservedEvents	Signale les événements observés dans une commande de AuditValue ou Notify.
Statistics	Signale, dans une commande Subtract ou Audit, des statistiques conservées au sujet d'une terminaison.
Topologie	Spécifie les directions des flux entre les terminaisons dans un contexte
Erreur	Contient un code d'erreur et éventuellement un texte d'erreur; il peut figurer dans les réponses aux commandes et dans les demandes de notification
NOTE – Le descripteur ModemDescriptor a été désapprouvé dans la Rec. H.248.1 version 1 (03/2002).	

6.2.5 Terminaison racine

Occasionnellement, une commande doit faire référence à la passerelle en tant qu'entité en soi plutôt qu'à une des terminaisons qu'elle contient. Un identificateur de terminaison spécial, Root, est réservé à cette fin. Des paquetages peuvent être définis au sujet d'une terminaison racine, qui peut donc avoir des propriétés, des événements, des signaux et des statistiques. En conséquence, l'identificateur de terminaison racine peut apparaître dans les messages suivants:

- une commande Modify – afin de modifier une propriété, d'envoyer un signal ou de fixer un événement;
- une commande Notify – afin de signaler un événement;
- une commande de renvoi AuditValue – afin d'examiner les valeurs des propriétés et des statistiques implémentées dans la terminaison racine;
- une commande AuditCapability – afin de déterminer les propriétés implémentées dans la terminaison racine;
- une commande ServiceChange – afin de déclarer la passerelle en service ou hors service.

Tout autre usage de l'identificateur de terminaison racine est une erreur. Code d'erreur 410 (L'identificateur incorrect) doit être renvoyé dans ces cas.

7 Commandes

Le protocole fournit des commandes pour manipuler les entités logiques du modèle de connexion du protocole, les contextes et les terminaisons. Ces commandes permettent un réglage au plus fin degré de granularité pris en charge par le protocole. Par exemple, des commandes permettent d'ajouter des terminaisons à un contexte, de modifier des terminaisons, de soustraire des terminaisons d'un contexte et d'analyser les propriétés de contextes ou de terminaisons. Les commandes permettent un contrôle complet des propriétés des contextes et des terminaisons, dont la spécification des événements qu'une terminaison doit signaler, la spécification des signaux/actions qui doivent être appliqués à une terminaison et la spécification de la topologie d'un contexte (qui entend/voit qui).

La plupart des commandes sont réservées à l'usage particulier du contrôleur MGC en tant qu'émetteur de commandes pour contrôler des passerelles MG en tant que récepteurs de commandes. Les exceptions sont les commandes Notify et ServiceChange, la première étant envoyée par une passerelle MG à un contrôleur MGC et la seconde pouvant être envoyée par une des deux entités. On trouvera ci-dessous une vue du paquetage des commandes, qui sont expliquées plus en détail dans le § 7.2.

1) **Add**: cette commande ajoute une terminaison à un contexte. Appliquée à la première terminaison d'un contexte, elle sert à créer un contexte.
2) **Modify**: cette commande modifie les propriétés, les événements et les signaux d'une terminaison.
3) **Subtract**: cette commande déconnecte une terminaison de son contexte et renvoie des statistiques sur la participation de cette terminaison à ce contexte. Appliquée à la dernière terminaison d'un contexte, elle sert à supprimer ce contexte.
4) **Move**: cette commande déplace atomiquement une terminaison vers un autre contexte.
5) **AuditValue**: cette commande renvoie les états actuels des propriétés, des événements, des signaux et des statistiques associés aux terminaisons.
6) **AuditCapabilities**: cette commande renvoie toutes les valeurs possibles des propriétés, des événements et des signaux de terminaisons, autorisées par la passerelle MG.

7) **Notify**: cette commande permet à la passerelle MG d'informer le contrôleur MGC de l'apparition d'événements dans cette passerelle.

8) **ServiceChange**: cette commande permet à la passerelle MG de signaler au contrôleur MGC qu'une terminaison ou un groupe de terminaisons est sur le point d'être mis hors service ou vient d'être remis en service. Cette commande est également utilisée par la passerelle MG pour annoncer sa disponibilité à un contrôleur MGC (enregistrement) et pour signaler au contrôleur MGC le redémarrage imminent ou récent de cette passerelle MG. Le contrôleur MGC peut annoncer un transfert à la passerelle MG en lui envoyant une commande ServiceChange. Le contrôleur MGC peut également utiliser cette commande pour demander à la passerelle MG de mettre en ou hors service une terminaison ou un groupe de terminaisons.

Ces commandes sont détaillées aux § 7.2.1 à 7.2.8.

7.1 Descripteurs

Les paramètres relatifs à une commande s'appellent des descripteurs. Un descripteur se compose d'un nom et d'une liste d'éléments dont certains peuvent avoir des valeurs. De nombreuses commandes partagent des descripteurs communs. Le présent paragraphe énumère ces descripteurs. Les descripteurs peuvent être fournis en tant que données de sortie d'une commande. Dans tout renvoi de contenu de descripteur de la sorte, un descripteur vide est représenté par son nom sans être accompagné d'aucune liste. Les paramètres et leur utilisation spécifique pour un type de commande donné sont décrits dans le paragraphe relatif à cette commande.

7.1.1 Spécification des paramètres

Les paramètres de commande sont structurés en un certain nombre de descripteurs. En général, le format alphanumérique des descripteurs est le suivant: DescriptorName=<someID>{parm=value, parm=value…}.

Les paramètres peuvent être entièrement spécifiés, surspécifiés ou sous-spécifiés:

1) les paramètres entièrement spécifiés ont une seule valeur non ambiguë, que l'émetteur de commande demande au récepteur de la commande d'utiliser pour le paramètre spécifié;

2) les paramètres sous-spécifiés utilisent la valeur "CHOOSE" afin de permettre au récepteur de la commande de choisir toute valeur qu'il peut prendre en charge;

3) les paramètres surspécifiés possèdent une liste de valeurs possibles dont l'ordre dans la liste correspond à l'ordre de sélection préféré par l'émetteur de la commande. Le récepteur de la commande choisit une seule valeur dans la liste offerte puis renvoie cette valeur à l'émetteur de la commande.

Si un descripteur obligatoire autre que le descripteur Audit est non spécifié (c'est-à-dire totalement absent) pour une commande, l'éventuel paquetage de valeurs antérieur de ce descripteur pour cette terminaison est conservé. Dans des commandes autres que Substract, un descripteur Audit manquant équivaut à un descripteur Audit vide. Le comportement de la passerelle MG au regard de paramètres non spécifiés dans un descripteur varie en fonction du descripteur considéré, comme indiqué dans les paragraphes qui suivent. Chaque fois qu'un paramètre est sous-spécifié ou surspécifié, le descripteur qui contient la valeur choisie par le récepteur est inclus en tant que donnée de sortie de la commande.

Chaque commande spécifie le paramètre TerminationId sur lequel elle opère. Cet identificateur TerminationId peut être "remplacé par un caractère générique". Si tel est le cas, l'effet doit être identique comme si la commande était répétée avec chacun des identificateurs de terminaisons concordantes.

7.1.2 Descripteur Modem

Le descripteur Modem spécifie le type et les paramètres d'un modem, éventuellement nécessaires pour utilisation lors d'une conversation en mode H.324 et en mode texte. Ce descripteur désigne les types de modem suivants: V.18, V.22, V.22 *bis*, V.32, V.32 *bis*, V.34, V.90, V.91, RNIS synchrone, avec possibilité d'extensions. Par défaut, aucun descripteur Modem n'est présent dans une terminaison.

L'emploi du descripteur ModemDescriptor est désapprouvé dans la Rec. UIT-T H.248.1 version 1 (03/2002) et dans les versions ultérieures. Cela veut dire que ce descripteur ne fera pas partie du contenu transmis, et, s'il est reçu, il sera soit ignoré soit traité en fonction de l'implémentation. Le type de modem est à spécifier comme attribut des flux de données dans les descripteurs LocalDescriptor et RemoteDescriptor.

7.1.3 Descripteur Mux

Dans les communications multimédias, un certain nombre de flux médias sont acheminés sur un certain nombre (éventuellement différent du précédent) de supports. Le descripteur Mux associe les médias et les supports. Il désigne les types de multiplex suivants:

- H.221;
- H.223;
- H.226;
- V.76;
- N × 64K;
- extensions possibles,

ainsi qu'un paquetage d'identificateurs de terminaison représentant les supports multiplexés, dans l'ordre. Par exemple:

```
Mux = H.221{ MyT3/1/2, MyT3/2/13, MyT3/3/6, MyT3/21/22}
```

Le type de multiplex N × 64K permet d'implémenter un service à N × 64K (par exemple, tel que défini par le débit de transfert des informations de la Rec. UIT-T Q.931 ou par la caractéristique du support de transmission de la Rec. UIT-T Q.763). Du côté du contexte, il prend en charge un seul flux de données à large bande. Lorsqu'une terminaison support est implicitement ajoutée à un contexte à la suite de la création d'une terminaison de multiplexage à N × 64K, le descripteur streamDescriptor de la terminaison support prendra les mêmes valeurs que le descripteur streamDescriptor défini pour la terminaison multiplex, mais la largeur de bande de la terminaison support sera de 64 kbit/s.

7.1.4 Descripteur Media

Le descripteur Media spécifie les paramètres de tous les flux médias. Ces paramètres sont structurés en deux descripteurs: un descripteur TerminationState qui spécifie les propriétés d'une terminaison qui ne dépendent pas du flux et un ou plusieurs descripteurs Stream dont chacun décrit un seul flux média.

Un flux est identifié par un identificateur Stream servant à associer ce flux au contexte auquel il appartient. Plusieurs flux sortant d'une terminaison doivent être synchronisés les uns avec les autres. A l'intérieur du descripteur Stream, il existe trois sous-descripteurs, désignés par LocalControl, Local et Remote. La relation entre ces descripteurs est donc la suivante:

```
TerminationState Descriptor
Stream descriptor
```

```
LocalControl descriptor
Local descriptor
Remote descriptor
```

Pour plus de commodité, il est possible d'inclure les sous-descripteurs LocalControl, Local ou Remote dans le descripteur Media sans inclure de descripteur Stream. Dans ce cas, l'on part du principe que l'identificateur de flux est 1**.**

7.1.5 Descripteur TerminationState

Le descripteur TerminationState contient la propriété ServiceStates, la propriété EventBufferControl et les propriétés d'une terminaison (définies dans des paquetages) qui ne sont pas spécifiques du flux.

La propriété ServiceStates décrit l'état global de la terminaison (non spécifique du flux). Une terminaison peut être dans l'un des états suivants: "test" (essai), "out of service" (hors service), ou "in service" (en service). L'état "test" indique que la terminaison est soumise à l'essai. L'état "out of service" indique que la terminaison ne peut pas être utilisée pour le trafic. L'état "in service" indique qu'une terminaison peut être utilisée ou est en cours d'utilisation pour le trafic normal. L'état par défaut est "en service".

Les valeurs attribuées aux propriétés peuvent être des valeurs simples (nombre entier/chaîne/énumération) ou des valeurs sous-spécifiées. Dans ce dernier cas, plusieurs valeurs sont fournies et la passerelle MG peut effectuer un choix entre les possibilités suivantes:

- Alternative Values (autres valeurs): plusieurs valeurs dans une liste dont l'une doit être choisie.
- Ranges (plages; étendues): valeurs minimale et maximale; toute valeur entre min et max doit être choisie, les valeurs des bornes non comprises.
- Greater Than/Less Than (supérieure/inférieure): la valeur doit être supérieure/inférieure à la valeur spécifiée.
- Caractère générique CHOOSE: la passerelle MG choisit parmi les valeurs autorisées pour la propriété.

La propriété EventBufferControl spécifie si les événements sont mis en mémoire tampon après la détection d'un événement dans le descripteur d'événements ou s'ils sont traités immédiatement. Pour les détails, se reporter au § 7.1.9.

7.1.6 Descripteur Stream

Un descripteur Stream spécifie les paramètres d'un flux bidirectionnel unique. Ces paramètres sont structurés en trois descripteurs: un qui contient les propriétés spécifiques à un flux et un pour les flux local et distant. Le descripteur Stream comporte un identificateur de flux qui désigne le flux. On crée les flux en spécifiant un nouvel identificateur StreamID sur une des terminaisons dans un contexte. On supprime un flux en réglant des descripteurs Local et Remote vides pour ce flux, les paramètres ReserveGroup et ReserveValue dans LocalControl étant réglés à "Faux" sur toutes les terminaisons dans le contexte qui prenait précédemment en charge ce flux.

Les identificateurs StreamID ont une signification locale entre le contrôleur MGC et la passerelle MG; ils sont attribués par le contrôleur MGC. Dans un contexte, l'identificateur StreamID sert à indiquer les flux médias qui sont interconnectés: les flux de même identificateur StreamID sont connectés.

Si une terminaison est déplacée d'un contexte dans un autre, l'effet sur le contexte d'arrivée de la terminaison déplacée est le même que si on ajoutait une nouvelle terminaison ayant les mêmes identificateurs StreamID que la terminaison déplacée.

7.1.7 Descripteur LocalControl

Le descripteur LocalControl contient la propriété Mode, les propriétés ReserveGroup et ReserveValue ainsi que les propriétés d'une terminaison (définies dans des paquetages) qui sont spécifiques au flux et présentent un intérêt entre la passerelle MG et le contrôleur MGC. Les valeurs des propriétés peuvent être spécifiées comme indiqué au § 7.1.1.

Les valeurs autorisées pour la propriété Mode sont: "send-only" (émission seulement), "receive-only" (réception seulement), "send/receive" (émission/réception), "inactive" (inactive) et "loop-back" (bouclage). Les valeurs "send" et "receive" se rapportent à l'intérieur d'un contexte si bien que, par exemple, un flux réglé sur "mode = sendOnly" ne transmettra pas au contexte le média reçu. La valeur par défaut de la propriété de mode est "inactive". Les signaux et les événements ne sont pas affectés par le paramètre de mode.

Les propriétés Reserve à valeurs booléennes d'une terminaison, ReserveValue et ReserveGroup, indiquent ce que la passerelle MG doit faire lorsqu'elle reçoit un descripteur local et distant.

Si la valeur d'une propriété Reserve est mise à "True" (Vrai), la passerelle MG DOIT réserver des ressources pour toutes les variantes spécifiées dans les descripteurs locaux et distants pour lesquelles elle a actuellement des ressources disponibles. Elle DOIT répondre avec les variantes pour lesquelles elle réserve des ressources. Si elle ne peut prendre en charge une quelconque de ces variantes, elle DOIT envoyer au contrôleur MGC une réponse qui contient des descripteurs locaux et distants vides. Si les médias commencent à s'écouler tandis que plus d'une variante est réservée, les paquets médias peuvent être envoyés ou reçus par l'une quelconque des variantes et doivent être traités, même si à un moment donné une seule variante à la fois peut être active.

Si la valeur d'une propriété Reserve est mise à "False" (Faux), la passerelle MG DOIT choisir une des variantes spécifiées dans le descripteur local (s'il est présent) et une des variantes spécifiées dans le descripteur distant (s'il est présent). Si la passerelle MG n'a pas encore réservé de ressources pour prendre en charge la variante choisie, elle DOIT le faire. Si, d'autre part, elle a déjà réservé des ressources pour la terminaison concernée (du fait d'un échange précédent où ReserveValue et ReserveGroup sont égaux à "True"), elle DOIT libérer les éventuelles ressources en excès qu'elle avait précédemment réservées. Enfin, la passerelle MG doit envoyer au contrôleur MGC une réponse contenant les variantes pour le descripteur local et distant qu'elle a choisi. Si la passerelle n'a pas les ressources suffisantes pour prendre en charge une quelconque des variantes spécifiées, elle DOIT répondre avec une erreur 510 (ressources insuffisantes).

La valeur par défaut de ReserveValue et ReserveGroup est "False" (Faux). Des renseignements complémentaires relatifs à l'utilisation des deux propriétés Reserve se trouvent au § 7.1.8.

Un nouveau réglage du descripteur LocalControl remplace entièrement le réglage précédent de ce descripteur dans la passerelle MG. Ainsi, pour conserver des informations provenant du réglage précédent, la passerelle MGC doit inclure ces informations dans le nouveau réglage. Si le contrôleur MGC souhaite supprimer une quelconque information du descripteur existant, il renvoie simplement le descripteur (dans une commande Modify), l'information indésirable en étant éliminée.

7.1.8 Descripteurs Local et Remote

Le contrôleur MGC utilise les descripteurs Local et Remote pour réserver et engager des ressources de la passerelle MG en vue du codage et décodage de média pour le ou les flux et terminaison auxquels elles s'appliquent. La passerelle inclut ces descripteurs dans sa réponse afin d'indiquer ce qu'elle est préparée à prendre effectivement en charge. La passerelle MG DOIT inclure des propriétés supplémentaires et leurs valeurs dans sa réponse si ces propriétés sont obligatoires bien que ne figurant pas dans les demandes faites par le contrôleur MGC (par exemple en spécifiant des paramètres de codage vidéo détaillés alors que le contrôleur MGC n'a spécifié que le type de charge utile).

Le descripteur "Local" se rapporte aux médias reçus par la passerelle MG et le descripteur "Remote" à ceux qu'elle envoie.

Lors du codage en mode texte du protocole, ces descripteurs se composent de descriptions de session telles que définies dans le protocole SDP (RFC 2327). Dans les descriptions de session envoyées par le contrôleur MGC à la passerelle MG, les exceptions suivantes à la syntaxe de RFC 2327 sont autorisées:

- les lignes "s = ", "t = " et "o = " sont facultatives;
- l'utilisation de CHOOSE est permise à la place d'une seule valeur de paramètre;
- l'utilisation de variantes est autorisée à la place d'une seule valeur de paramètre.

Un descripteur Stream spécifie un seul flux média bidirectionnel et une unique description de session NE DOIT donc PAS contenir plus d'une description de média (ligne "m = "). Un descripteur de session peut contenir des descriptions de session supplémentaires en tant que solutions de remplacement. Chaque flux média d'une terminaison doit figurer dans un descripteur Stream distinct. Lorsque plusieurs descriptions de session sont fournies dans un même descripteur, les lignes "v = " sont requises en tant que délimiteurs; sinon, elles sont facultatives dans les descriptions de session envoyées à la passerelle MG. Les implémentations doivent accepter des descriptions de session entièrement conformes au document RFC 2327. Lors du codage en mode binaire du protocole, le descripteur se compose de groupes de paramètres (paires de valeurs d'étiquette) comme spécifié dans l'Annexe C. Chacun de ces groupes peut contenir les paramètres d'une description de session.

La sémantique des descripteurs Local et Remote est spécifiée en détails ci-après. La spécification comporte deux parties. La première partie spécifie l'interprétation du contenu du descripteur. La seconde partie spécifie les actions que la passerelle MG doit entreprendre à la réception des descripteurs Local et Remote. Ces actions dépendent des valeurs des propriétés ReserveValue et ReserveGroup du descripteur LocalControl.

Chacun des descripteurs Local et Remote (ou tous les deux) peut être:

- non spécifié (c'est-à-dire absent);
- vide;
- sous-spécifié par le biais de l'utilisation de CHOOSE dans une valeur de propriété;
- complètement spécifié;
- surspécifié par le biais de la présentation de plusieurs groupes de propriétés et de plusieurs valeurs de propriétés possibles dans un ou plusieurs de ces groupes.

Lorsqu'ils sont transmis du contrôleur MGC à la passerelle MG, les descripteurs sont interprétés conformément aux règles indiquées dans le § 7.1.1, avec les commentaires supplémentaires suivants dans un but de clarification:

a) un descripteur Local ou Remote non spécifié est considéré comme étant un paramètre obligatoire qui manque. Il exige que la passerelle MG utilise la dernière valeur spécifiée pour ce descripteur. Il est possible qu'il n'y ait pas de valeur précédemment spécifiée, auquel cas le descripteur concerné est ignoré dans le traitement ultérieur de la commande;

b) un descripteur Local (Remote) vide contenu dans un message provenant du contrôleur MGC signifie une demande de libérer les éventuelles ressources réservées pour le flux média qui ont été reçues (envoyées);

c) si plusieurs groupes de propriétés sont présents dans un descripteur Local ou Remote ou s'il y a plusieurs valeurs dans un groupe, l'ordre de préférence est décroissant;

d) les propriétés sous-spécifiées ou surspécifiées contenues dans un groupe de propriétés envoyées par le contrôleur MGC constituent des demandes à la passerelle MG de choisir une ou plusieurs valeurs qu'elle peut prendre en charge pour chacune de ces propriétés. Dans le cas d'une propriété surspécifiée, la liste des valeurs est dans l'ordre décroissant de préférence.

Les actions subséquentes soumises aux règles précédentes dépendent des valeurs des propriétés ReserveValue et ReserveGroup du descripteur LocalControl.

Si la valeur de ReserveGroup est "True" (Vrai), la passerelle MG réserve les ressources requises pour prendre en charge l'une quelconque des variantes de groupes de propriétés qu'elle peut prendre actuellement en charge. Si la valeur de ReserveValue est "True" (Vrai), la passerelle MG réserve les ressources requises pour prendre en charge l'une quelconque des variantes de valeurs de propriétés qu'elle peut prendre actuellement en charge.

NOTE – Si un descripteur Local ou Remote contient plusieurs groupes de propriétés et si la valeur de ReserveGroup est "True" (Vrai), il est demandé à la passerelle MG de réserver des ressources de manière à ce qu'elle puisse décoder ou coder le flux média conformément à l'une quelconque des variantes. Par exemple, si le descripteur Local contient deux groupes de propriétés, l'un spécifiant l'audio de la loi A de G.711 en mode paquet et l'autre l'audio de G.723.1, la passerelle MG réserve des ressources de manière à pouvoir décoder un flux audio codé dans le format loi A de G.711 ou dans le format G.723.1 La passerelle n'a pas à réserver des ressources pour décoder simultanément deux flux audio, l'un codé selon le format loi A de G.711 et l'autre dans le format G.723.1. L'utilisation de ReserveValue est analogue.

Si la valeur de ReserveGroup est "True" ou si celle de ReserveValue est "True", les règles ci-après s'appliquent:

- si la passerelle MG n'a pas de ressources suffisantes pour prendre en charge toutes les variantes demandées par le contrôleur MGC et si celui-ci a demandé des ressources dans chacun des descripteurs Local et Remote, il convient que la passerelle MG réserve des ressources pour prendre en charge au moins une variante dans chaque descripteur Local et Remote;
- si la passerelle MG n'a pas de ressources suffisantes pour prendre en charge au moins une variante dans un descripteur Local (Remote) reçu du contrôleur MGC, elle doit renvoyer un descripteur Local (Remote) vide comme réponse;
- dans sa réponse au contrôleur MGC, lorsque celui-ci avait inclus des descripteurs Local et Remote, la passerelle MG DOIT inclure des descripteurs Local et Remote pour tous les groupes de propriétés et toutes les valeurs de propriétés pour lesquels elle avait réservé des ressources. Si la passerelle est incapable de prendre en charge au moins l'une des alternatives contenues dans le descripteur Local (Remote) reçu du contrôleur MGC, elle DOIT renvoyer un descripteur Local (Remote) vide;
- si la propriété Mode du descripteur LocalControl est RecvOnly, SendRecv ou LoopBack, la passerelle MG doit être préparée pour recevoir des médias codés selon l'une quelconque des variantes incluses dans sa réponse au contrôleur MGC.

Si la valeur de ReserveGroup est "False" (Faux) et si celle de ReserveValue est "False" (Faux), IL CONVIENT que la passerelle MG applique les règles ci-après pour délimiter Local et Remote chacun à une seule variante:

- la passerelle MG choisit la première variante dans le descripteur Local pour laquelle elle est à même de prendre en charge au moins une variante dans le descripteur Remote;
- si la passerelle MG est incapable de prendre en charge au moins une variante contenue dans Local et une contenue dans Remote, elle renvoie l'erreur 510 (Ressources insuffisantes);
- la passerelle renvoie la variante qu'elle a choisie dans Local et dans Remote.

Un nouveau réglage d'un descripteur Local ou Remote remplace entièrement le réglage précédent de ce descripteur dans la passerelle MG. Ainsi, pour conserver des informations provenant du réglage précédent, le contrôleur MGC doit inclure ces informations dans le nouveau réglage. Si le contrôleur MGC souhaite supprimer une quelconque information du descripteur existant, il renvoie simplement le descripteur (dans une commande Modify), l'information indésirable en étant éliminée.

7.1.9 Descripteur d'événements

Le paramètre EventsDescriptor contient un identificateur de demande et une liste d'événements que la passerelle MG est appelée à détecter et à signaler. L'identificateur de demande (RequestIdentifier) est utilisé pour associer la demande aux notifications que celle-ci peut déclencher. Les événements demandés sont, par exemple, des tonalités de télécopie, des résultats d'essai de continuité et des transitions de raccrochage-décrochage. Le paramètre RequestIdentifier est omis si le paramètre EventsDescritor est vide (c'est-à-dire aucun événement n'est spécifié).

Chaque événement indiqué dans le descripteur Events contient le nom de l'événement, un identificateur de flux facultatif, un fanion KeepActive (maintenir en activité) facultatif et des paramètres facultatifs. Le nom d'événement se compose d'un nom de paquetage (dans lequel l'événement est défini) et d'un identificateur d'événement. Le caractère générique ALL peut être utilisé pour l'identificateur d'événement afin d'indiquer que tous les événements issus du paquetage spécifié doivent être détectés. L'identificateur de flux par défaut est 0, indiquant que l'événement à détecter n'est pas associé à un flux média particulier. Les événements peuvent avoir des paramètres, ce qui permet à une même description d'événement d'avoir une certaine variation de sens, ce qui évite d'avoir à créer de grands nombres d'événements individuels. Les autres paramètres d'événement sont définis dans le paquetage.

Lorsqu'un événement de fin de script de numérotation est présent dans l'identificateur EventsDescriptor, de manière explicite ou implicite, le paramètre EventDM est utilisé pour acheminer soit le nom soit la valeur du script de numérotation associé. Voir 7.1.14 pour plus de détails.

Lorsqu'un événement est traité en fonction du contenu d'un descripteur EventsDescriptor actif et qu'il s'avère être présent dans ce descripteur ("reconnu"), l'action par défaut de la passerelle MG consiste à envoyer une commande Notify au contrôleur MGC. La notification peut être différée si l'événement est absorbé dans la chaîne de numérotation courante d'un script de numérotation actif (voir § 7.1.14). Toute autre action est l'objet d'une étude plus approfondie. De plus, la reconnaissance d'événement peut provoquer l'arrêt de signaux actifs ou le remplacement du descripteur actuel Events et/ou Signals, conformément à la description donnée à la fin du présent paragraphe. A moins que le descripteur EventsDescriptor ne soit remplacé par un autre descripteur, il reste actif après qu'un événement a été reconnu.

Si la valeur dans la propriété EventBufferControl est égale à LockStep, à la suite de la détection d'un tel événement, le traitement normal des événements est suspendu. Tout événement détecté subséquemment et présent dans le descripteur EventBuffer (mémoire tampon d'événements) est ajouté à la fin de la mémoire tampon d'événements (une file d'attente FIFO – premier arrivé, premier servi) ainsi que l'heure à laquelle il a été détecté. La passerelle DOIT attendre le chargement d’un nouveau descripteur d'événements. Un nouveau descripteur d'événement (EventsDescriptor) peut être chargé soit en tant que résultat de la réception d'une commande avec un nouveau EventsDescriptor soit en activant le descripteur Events incorporé.

Si EventBufferControl est égal à "Off" (inactif), la passerelle MG poursuit le traitement en s'appuyant sur le descripteur Events actif.

Dans le cas d'activation d'un descripteur Events incorporé, la passerelle MG poursuit le traitement des événements en s'appuyant sur le descripteur Events nouvellement activé.

NOTE 1 – Pour des besoins de traitement de mémoire tampon d'événements, l'activation d'un descripteur Events incorporé équivaut à la réception d'un nouveau descripteur Events.

Lorsque la passerelle MG reçoit une commande avec un nouveau descripteur d'événements (EventsDescriptor), sa mémoire tampon d'événements peut déjà contenir un ou plusieurs événements. La valeur de EventBufferControl détermine alors la manière dont la passerelle MG traite ces événements placés en mémoire tampon.

Cas 1

Si EventBufferControl est égal à LockStep et si la passerelle MG reçoit un nouveau descripteur Events, elle vérifie la mémoire tampon d'événements FIFO et exécute les actions suivantes:

1) si la mémoire tampon d'événements est vide, la passerelle MG attend la détection d'événements fondés sur le nouveau descripteur Events;

2) si la mémoire tampon d'événements n'est pas vide, la passerelle MG traite la file d'attente FIFO en commençant par le premier événement:

   a) si l'événement dans la file d'attente figure parmi les événements énumérés dans le nouveau descripteur Events, la passerelle MG agit sur cet événement en le retirant de la mémoire tampon d'événements. L'horodatage de la commande Notify doit être l'heure de la détection effective de l'événement. La passerelle attend ensuite un nouveau descripteur Events. Pendant cette attente, tous les événements détectés qui apparaissent dans le descripteur EventsBuffer sont placés dans la mémoire tampon d'événements. Lorsqu'un nouveau descripteur Events est reçu, le traitement des événements se répète à partir de l'étape 1;

   b) si l'événement n'est pas contenu dans le nouveau descripteur Events, la passerelle MG DOIT l'ignorer et recommencer depuis l'étape 1).

Cas 2

Si EventBufferControl est égal à "Off" (inactif) et si la passerelle MG reçoit un nouveau descripteur Events, elle traite les nouveaux événements contenus dans le nouveau descripteur Events.

Si la passerelle MG reçoit une commande lui demandant de fixer la valeur de EventBufferControl à "Off" (inactif), tous les événements contenus dans la mémoire tampon d'événements DOIVENT être ignorés.

La passerelle MG peut rapporter plusieurs événements dans une seule transaction tant que cela ne retarde pas inutilement le rapport des événements individuels.

Pour les procédures relatives à la transmission de la commande Notify, se reporter à l'annexe appropriée ou aux Recommandations H.248.x concernant des considérations de transport spécifiques.

La valeur par défaut de la propriété EventBufferControl est "Off" (inactif).

NOTE 2 – Etant donné que la propriété EventBufferControl est contenue dans le descripteur TerminationState (descripteur d'état de terminaison), la passerelle pourrait recevoir une commande qui modifie la propriété EventBufferControl et n'inclut pas un descripteur Events.

Normalement, la reconnaissance d'un événement arrête tous les signaux actifs éventuels. Lorsque KeepActive est spécifié dans l'événement, la passerelle MG ne doit pas interrompre d'éventuels signaux qui étaient actifs dans la terminaison où l'événement a été détecté.

Un événement peut comporter un descripteur de signal incorporé et un descripteur d'événement incorporé qui, s'ils sont présents lors de la reconnaissance de l'événement, remplacent le descripteur de signaux ou d'événements actuel. Il est possible, par exemple, de spécifier que le signal de tonalité d'invitation à numéroter sera produit lors de la reconnaissance d'un événement de décrochage ou que le signal de tonalité d'invitation à numéroter soit interrompu si un chiffre est reconnu. Un

contrôleur MGC ne doit pas envoyer de descripteurs d'événements contenant un événement qui, à la fois, est marqué et contient un descripteur de signaux incorporé.

Un seul niveau d'incorporation est autorisé. Un descripteur d'événements incorporé NE DOIT PAS contenir un autre descripteur d'événement incorporé mais il PEUT contenir un descripteur de signaux incorporé.

Un descripteur d'événement reçu par une passerelle MG remplace tout descripteur d'événement antérieur. La notification d'événement en cours doit se terminer et les événements détectés après l'exécution de la commande contenant le nouveau descripteur d'événement doivent être traités conformément au nouveau descripteur d'événements.

Un descripteur d'événements vide désactive toute reconnaissance ou tout rapport d'événement. Il vide la mémoire tampon EventBuffer et désactive toute accumulation d'événements en mode LockStep: les seuls événements rapportés seront ceux qui se produisent tandis que le descripteur EventsDescriptor est actif. Si le descripteur EventsDescriptor est activé pendant que la terminaison fonctionne en mode LockStep, la mémoire tampon d'événements est immédiatement vidée.

7.1.10 Descripteur EventBuffer (mémoire tampon d'événements)

Le descripteur EventBuffer contient une liste d'événements, avec leurs paramètres le cas échéant, que la passerelle MG est appelée à détecter et mémoriser en tampon lorsque la propriété EventBufferControl est égale à "LockStep" (se reporter au § 7.1.9).

7.1.11 Descripteur de signaux

Les signaux sont des médias produits par une passerelle MG, tels que des tonalités et des annonces, ainsi que des signaux liés au support tels que ceux qui sont destinés au crochet commutateur. Des signaux plus complexes peuvent comporter une suite de tels signaux simples, entrecoupée de signaux médias ou liés au support, et conditionnée par la réception et l'analyse de ceux-ci. Des exemples en sont le renvoi en écho de données reçues comme dans le paquetage des essais de continuité. Les signaux peuvent aussi demander que le contenu média soit préparé pour des signaux ultérieurs.

Un descripteur Signals est un paramètre qui contient le paquetage des signaux que la passerelle MG est appelée à appliquer à une terminaison. Un descripteur Signals contient un certain nombre de signaux et de listes séquentielles de signaux. Un descripteur Signals peut contenir un nombre égal à zéro de signaux et de listes séquentielles de signaux. La prise en charge de listes séquentielles de signaux est facultative.

Les signaux sont définis dans des paquetages. Le nom d'un signal doit être celui du paquetage dans lequel il est défini, plus un identificateur de signal. Aucun caractère générique ne doit être utilisé comme identificateur de signal. Les signaux qui sont contenus dans un descripteur de signaux possèdent un paramètre facultatif d'identificateur de flux (dont la valeur par défaut est 0, ce qui indique que le signal n'est associé à aucun flux média particulier), un type facultatif de signal (voir ci-dessous), une durée facultative et éventuellement des paramètres définis dans le paquetage qui définit le signal. Cela permet à un même signal d'avoir une certaine variation de sens ce qui évite d'avoir à créer de grands nombres de signaux individuels.

Enfin, le paramètre facultatif "notifyCompletion" permet à un contrôleur MGC d'indiquer son souhait d'être avisé lorsque le signal achève son exécution. Les cas possibles sont les suivants: expiration de la temporisation du signal (ou sinon achèvement de lui-même), interruption du signal par un événement, arrêt du signal par le remplacement d'un descripteur Signals ou arrêt ou non-lancement pour d'autres raisons. Si le paramètre notifyCompletion ne figure pas dans un descripteur Signals, une notification n'est générée que si le signal a été arrêté ou s'il n'a jamais été lancé pour d'autres raisons. Pour qu'il soit rapporté, l'événement de fin de signal (voir § E.1.2) doit être activé dans le descripteur Events actuellement actif.

La durée est une valeur entière exprimée en centièmes de seconde.

Il existe trois types de signaux:

- signal commuté: le signal dure jusqu'à ce qu'il soit interrompu;
- signal temporisé: le signal dure jusqu'à ce qu'il soit interrompu ou jusqu'à ce qu'un intervalle de temps spécifique se soit écoulé;
- signal bref: le signal s'interrompra de lui-même à moins qu'un nouveau descripteur Signals ne soit appliqué pour l'interrompre plus tôt; aucune valeur de temporisation n'est nécessaire.

Si le type par défaut d'un signal autre que le type TO est supplanté par celui-ci dans le descripteur Signals, le paramètre de durée doit être présent.

Si le type du signal est spécifié dans le descripteur de signaux, il supplante le type de signal par défaut (se reporter au § 12.1.4). Si une durée est spécifiée pour un signal commuté, elle DOIT être ignorée.

Une liste séquentielle de signaux se compose d'un identificateur de liste de signaux et d'une séquence de signaux à exécuter consécutivement. Seul l'élément final de la séquence de signaux, dans une telle liste, peut être un signal commuté. La durée d'une liste séquentielle de signaux est la somme des durées des signaux qu'elle contient.

Plusieurs signaux et listes séquentielles de signaux peuvent être exécutés simultanément dans le même descripteur de signaux.

Les signaux sont définis comme se propageant de la terminaison vers l'extérieur du contexte, sauf spécification contraire dans un paquetage. Lorsque le même signal est appliqué à plusieurs terminaisons contenues dans une même transaction, la passerelle MG doit envisager d'utiliser la même ressource pour produire ces signaux.

La production d'un signal dans une terminaison est arrêtée par l'application d'un nouveau descripteur de signaux ou par la détection d'un événement dans cette terminaison (se reporter au 7.1.9).

Un nouveau descripteur de signaux remplace tout descripteur de signaux existant. Tous les signaux appliqués à la terminaison qui ne sont pas indiqués dans le descripteur de remplacement doivent être arrêtés et de nouveaux signaux doivent être appliqués, sauf dans les conditions suivantes. Les signaux présents dans le descripteur de remplacement et contenant le fanion KeepActive doivent se poursuivre s'ils sont en cours d'exécution et n'ont pas encore fini. Si un signal qui n'est pas en cours d'exécution et qui contient le fanion KeepActive est contenu dans un descripteur de signaux de remplacement, ce signal DOIT être ignoré. Si le descripteur de remplacement contient une liste séquentielle de signaux contenant le même identificateur que le descripteur existant,

- le type de signal et la séquence de signaux dans la liste séquentielle figurant dans le descripteur de remplacement doivent être ignorés;
- l'exécution des signaux contenus dans la liste séquentielle du descripteur existant ne doit pas être interrompue.

7.1.12 Descripteur Audit

Le descripteur Audit spécifie les informations à auditer. Il spécifie la liste de descripteurs et/ou les propriétés distinctes à retourner. Il peut être utilisé dans n'importe quelle commande afin de forcer le renvoi d'un descripteur quelconque contenant les valeurs courantes de ses propriétés, événements, signaux et statistiques même si ce descripteur n'était pas présent dans la commande ou ne possède aucun paramètre sous-spécifié. Les éléments pouvant être contenus dans le descripteur Audit sont les suivants:

Modem (désapprouvé, voir 7.1.2)
Mux
Events
Media
Signals
ObservedEvents
DigitMap
Statistics
Packages
EventBuffer
Eléments analysés distincts: Propriétés des médias Evénements Tampon d'événements Signaux, listes de signaux Scripts de numérotation Statistiques Paquetages

La commande Audit peut être vide: dans ce cas, aucun descripteur n'est renvoyé. Cela est utile dans la commande Subtract, afin d'empêcher le renvoi de statistiques, surtout en cas d'utilisation d'un caractère générique.

7.1.13 Descripteur ServiceChange (changement de service)

Ce descripteur contient les paramètres suivants:

- ServiceChangeMethod
- ServiceChangeReason
- ServiceChangeAddress
- ServiceChangeDelay
- ServiceChangeProfile
- ServiceChangeVersion
- ServiceChangeMGCId
- TimeStamp
- Extension

Se reporter au § 7.2.8.

7.1.14 Descripteur DigitMap (script de numérotation)

7.1.14.1 Définition, création, modification et suppression d'un script de numérotation

Un script de numérotation est un plan de numérotation résidant dans la passerelle MG, qui sert à détecter et à signaler les événements de numérotation reçus au sujet d'une terminaison. Le descripteur DigitMap contient le nom d'un script d'instructions et le script à attribuer. Un script peut être préenregistré dans la passerelle MG par une action de gestion et être répertorié par son nom dans un descripteur d'événements; il peut également être défini dynamiquement puis être répertorié par son nom ou bien le script de numérotation lui-même peut être spécifié dans le descripteur Events. Il est admissible qu'un événement de fin de script de numérotation contenu dans un descripteur d'événements se rapporte par nom à un script de numérotation qui est défini par un descripteur de scripts de numérotation contenu dans la même commande, indépendamment de l'ordre dans lequel ces descripteurs respectifs ont été transmis.

Les scripts de numérotation définis dans un descripteur DigitMap peuvent apparaître dans une quelconque des commandes normales de manipulation de terminaison du protocole. Une fois définie, un script de numérotation peut être utilisé dans toutes les terminaisons spécifiées par l'identificateur de terminaison (éventuellement remplacé par un caractère générique) contenu dans une telle commande. Les descripteurs DigitMap définis dans une terminaison racine sont globaux et on peut les utiliser dans toutes les terminaisons contenues dans la passerelle MG, à la condition qu'un descripteur DigitMap portant le même nom n'ait pas été défini dans la terminaison considérée. Lorsqu'un script de numérotation est défini dynamiquement dans un descripteur DigitMap:

- un nouveau script de numérotation est créé par la spécification d'un nom qui n'est pas encore défini mais dont la valeur doit être présente;
- une valeur de script de numérotation est mise à jour par fourniture d'une nouvelle valeur pour un nom déjà défini; les terminaisons utilisant actuellement le script de numérotation doivent continuer à utiliser l'ancienne définition; les descripteurs Events subséquents qui utilisent le nom, y compris tout descripteur Events dans la commande contenant le descripteur DigitMap, doivent utiliser le nouveau nom;
- un script de numérotation est supprimé par fourniture d'une valeur vide pour un nom déjà défini. Les terminaisons utilisant actuellement le script de numérotation doivent continuer à utiliser l'ancienne définition.

7.1.14.2 Temporisateurs de script de numérotation

Le paquetage des chiffres conformes à un script de numérotation peut être protégé par trois temporisateurs, à savoir un temporisateur de départ (T), un temporisateur court (S) et un temporisateur long (L), comme suit:

1) le temporisateur de départ (T) est utilisé avant que des chiffres aient été composés. Si la valeur du temporisateur de départ est supplantée par une valeur fixée à zéro (T = 0), le temporisateur de départ doit être désactivé. Cela implique que la passerelle MG attendra indéfiniment les chiffres.

2) si la passerelle MG peut déterminer qu'au moins un chiffre de plus est nécessaire dans une chaîne de chiffres pour que celle-ci corresponde à l'une des structures autorisées dans le script de numérotation, la valeur du temporisateur entre chiffres doit être réglée sur une longue (L) durée (par exemple 16 s);

3) si la chaîne de chiffres correspond à un des motifs contenus dans un script de numérotation mais qu'il est possible que d'autres chiffres soient reçus pour aboutir à une correspondance avec un motif différent, la passerelle doit appliquer un temporisateur court (S) et attendre d'autres chiffres, au lieu de rapporter immédiatement la correspondance.

Les temporisateurs sont des paramètres configurables en fonction d'un script de numérotation. Les valeurs par défaut de ces temporisateurs devraient être fournies au niveau de la passerelle MG, mais elles peuvent être supplantées par des valeurs spécifiées dans le script de numérotation.

7.1.14.3 Syntaxe de script de numérotation

La syntaxe formelle du script de numérotation est décrite par la règle DigitMap qui se trouve dans la description de syntaxe formelle du protocole (se reporter aux Annexes A et B). Conformément à cette syntaxe, un script de numérotation est défini soit par une chaîne soit par une liste de chaînes. Chaque chaîne de la liste est une séquence d'événements en variante, spécifiée soit sous forme d'une séquence de symboles de script de numérotation, soit sous forme d'une expression normale de symboles de script de numérotation. Les symboles de script de numérotation, chiffres de "0" à "9" et lettres de "A" à une valeur maximale qui dépend du système de signalisation concerné mais qui ne dépasse pas "K", correspondent aux événements spécifiés dans un paquetage qui a été désigné dans le descripteur d'événements sur la terminaison à laquelle le script est actuellement appliqué. (Le mappage entre événements et symboles de script de numérotation est défini dans la documentation des paquetages associés aux systèmes de signalisation en mode voie par voie tels que DTMF, MF ou R2. Les chiffres de "0" à "9" DOIVENT être mappés aux événements chiffres correspondants contenus dans le système de signalisation concerné. Il convient que les lettres soient attribuées de façon logique, afin de faciliter l'utilisation de la notation de gamme pour d'autres événements.)

La lettre "x" est utilisée comme caractère générique, désignant n'importe quel événement correspondant aux symboles dans la gamme de "0" à "9". La chaîne peut également contenir des gammes explicites et, d'une manière plus générale, des jeux de symboles explicites, désignant d'autres événements dont chacun satisfait à cette position dans le script de numérotation. Enfin le symbole point "." représente zéro ou davantage de répétitions du sélecteur d'événements (événement, gamme d'événements, jeu d'événements en variante ou caractère générique) qui le précède. Comme conséquence de la troisième règle de temporisation ci-dessus, la temporisation interévénements tout en correspondant à un symbole point terminal utilise par défaut le temporisateur court.

En plus de ces symboles d'événement, la chaîne peut contenir les spécificateurs de temporisation inter-événements "S" et "L" ainsi que le modificateur de durée "Z". "S" et "L" indiquent respectivement qu'il convient que la passerelle MG utilise le temporisateur court (S) ou le temporisateur long (L) pour des événements subséquents, annulant ainsi les règles décrites précédemment. Si un spécificateur de temporisation explicite est en vigueur dans une séquence d'événements en variante, mais qu'aucun autre n'est fourni dans les autres variantes candidates, on doit utiliser la valeur de temporisation fixée par le spécificateur explicite. Si toutes les séquences comportant des contrôles de temporisation explicites sont sorties de le paquetage candidat, la temporisation reprend les règles par défaut fournies précédemment. Enfin, le temporisateur long sera employé si des spécificateurs de temporisation sont en conflit et en vigueur dans des séquences en variante différentes.

La lettre "Z" désigne un événement de longue durée: placée en face de symbole(s) désignant un ou des événements qui satisfont à la position de chiffre donnée, elle indique que la position n'est satisfaite que si la durée de l'événement ne dépasse pas le seuil de longue durée. La valeur de ce seuil est supposée fournie dans la passerelle MG, mais elle peut, à l'instar des temporisateurs T, L et S, être supplantée par une autre valeur spécifiée dans le script de numérotation.

7.1.14.4 Evénement d'achèvement de script de numérotation

Un script de numérotation est actif tant que le descripteur d'événement qui l'a invoqué est actif et qu'il n'a pas été achevé. Un script de numérotation s'achève

- lorsqu'une temporisation a expiré;

- lorsqu'il y a eu correspondance avec une séquence d'événements en variante et qu'il ne peut pas y avoir de correspondance avec une autre séquence dans le script de numérotation par la détection d'un événement supplémentaire (correspondance non ambiguë);
- lorsqu'un événement a été détecté de telle manière qu'une correspondance avec une séquence complète d'événements en variante contenue dans le script de numérotation soit impossible, quels que soient les événements supplémentaires reçus.

Après l'achèvement, il doit être créé un événement d'achèvement de script de numérotation tel que défini dans le paquetage qui fournit les événements en cours de mappage dans le script. A ce moment-là, le script de numérotation est désactivé. Les événements subséquents du paquetage sont traités comme suit:

- si la propriété EventBufferControl est activée (ON), les événements de numérotation suivants sont traités de la même manière que les autres événements;
- si la propriété EventBufferControl est désactivée (OFF), que le descripteur EventsDescriptor n'a pas été modifié et que les événements de numérotation distincts ne sont pas activés dans ce descripteur, la mise en mémoire tampon de la numérotation sera entamée. Cette mise en mémoire se poursuivra jusqu'à ce que le temps imparti, spécifié dans l'événement d'achèvement du script de numérotation initial, soit expiré ou jusqu'à ce que le descripteur EventsDescriptor actif soit remplacé.

  La mémoire tampon de numérotation prendra la forme logique d'une chaîne de numérotation qui comporte les caractères numériques figurant dans le script de numérotation, éventuellement précédés de la lettre "Z". Le seuil de la durée de la tonalité employée pour identifier les événements de longue durée devra être le même que celui qui est utilisé pour le script de numérotation achevé en dernier lieu.

  Le temps de la mise en mémoire tampon est ramené à zéro par défaut (pas de mise en mémoire), à moins qu'une autre valeur ne soit explicitement indiquée dans l'événement d'achèvement du script de numérotation. Si la mise en mémoire s'arrête en raison de l'expiration du temporisateur de cette mise en mémoire, la mémoire tampon de numérotation est vidée de son contenu.

  Si la mise en mémoire tampon a été arrêtée par un nouveau descripteur EventsDescriptor, alors, si ce descripteur contient un nouvel événement d'achèvement du script de numérotation provenant du même paquetage que le précédent, les chiffres mis en mémoire sont traités conformément au script de numérotation comme décrit ci-après. Les chiffres mis en mémoire non utilisés par le nouveau script de numérotation sont traités comme s'ils avaient été observés après l'achèvement du script.

  Si, au lieu de cela, le nouveau descripteur EventsDescriptor autorise le rapport des événements de numérotation distincts, l'ensemble entier de chiffres mis en mémoire tampon sera immédiatement traité, les événements applicables seront signalés, et la mémoire tampon sera vidée.

  Finalement, si le nouveau descripteur EventsDescriptor n'active dans le paquetage concerné ni un événement d'achèvement du script de numérotation ni le rapport des événements de numérotation distincts, la mémoire tampon est vidée de son contenu et la mise en mémoire tampon s'achève.

7.1.14.5 Procédures de script de numérotation

En attendant l'achèvement, les événements subséquents doivent être traités selon les règles ci-après:

1) la chaîne "current dial string" (chaîne de numérotation actuelle), variable interne, est initialement vide. Le paquetage de séquences d'événements en variante candidates comprend toutes les variantes spécifiées dans le script de numérotation;

2) à chaque étape, si les chiffres mis en mémoire tampon sont disponibles, le plus ancien chiffre [accompagné éventuellement d'un qualificateur (Z) de numérotation longue] est écarté de la mémoire tampon et l'étape suivante du traitement est abordée, comme si l'événement de numérotation venait d'être observé. Sinon un temporisateur est réglé pour attendre l'événement suivant, en s'appuyant soit sur les règles de temporisation par défaut fournies ci-dessus, soit sur la temporisation explicite spécifiée dans une ou plusieurs séquences d'événements en variante. Si la temporisation expire et si un membre de le paquetage candidat de variantes est pleinement satisfait, un achèvement de temporisation avec correspondance totale est rapporté. Si la temporisation expire et si une partie d'une variante candidate est satisfaite ou si aucune ne l'est, un achèvement de temporisation avec correspondance partielle est rapporté. Dans les deux cas, si l'événement d'achèvement du script de numérotation autorise le rapport détaillé de l'expiration, la chaîne de numérotation signalée se terminera par les lettres "L", "S" ou "T", suivant le cas;

3) si un événement est détecté avant l'expiration de la temporisation, il est mappé à un symbole de chaîne de chiffres et ajouté provisoirement à la fin de la chaîne de numérotation en cours. La durée de l'événement (longue ou non) est notée si et seulement si c'est pertinent dans la position de symbole actuel (car au moins une des séquences d'événement en variante candidate contient le modificateur "Z" en cette position dans la séquence);

4) la chaîne de numérotation actuelle est comparée aux séquences d'événements en variante candidates. Si et seulement s'il y a correspondance avec une séquence attendant un événement de longue durée en cette position (c'est-à-dire que l'événement avait une longue durée et a satisfait à la spécification relative à cette position), toutes les séquences d'événements en variante qui ne spécifient pas un événement de longue durée en cette position sont ignorées et la chaîne de numérotation actuelle est modifiée par insertion de la lettre "Z" en face de l'événement représentant le dernier événement. Toute séquence attendant un événement de longue durée en cette position mais ne correspondant pas à l'événement observé est ignorée de l'ensemble candidat. Si, après application des règles ci-dessus, il reste dans l'ensemble candidat des séquences d'événements en variante qui ne spécifient pas un événement de longue durée dans la position donnée, la durée observée des événements est traitée comme n'étant pas pertinente dans l'évaluation de leurs correspondances;

5) s'il reste exactement un candidat et qu'il présente une correspondance complète, un événement d'achèvement est généré qui indique une correspondance non ambiguë. S'il ne reste aucun candidat, le dernier événement est retiré de la chaîne de numérotation actuelle et un événement d'achèvement est généré qui indique une correspondance totale si un candidat provenant de l'étape précédente avait été complètement satisfait avant la détection du dernier événement ou, autrement, il indique une correspondance partielle. L'événement qui a été retiré de la chaîne de numérotation actuelle est ensuite signalé comme un événement distinct, mis en mémoire tampon, ou écarté suivant les règles décrites dans le précédent paragraphe. Cette déclaration est libellée comme suit:

   a) l'événement d'achèvement du script de numérotation peut indiquer que l'événement écarté doit être signalé comme paramètre de l'événement d'achèvement. Ceci se fait indépendamment du traitement ultérieur de l'événement de numérotation;

   b) l'événement d'achèvement du script de numérotation peut indiquer que le chiffre supplémentaire doit être écarté. Dans ce cas, cela se fait immédiatement. Toute mise en mémoire tampon ou tout autre traitement ne s'applique qu'aux événements suivants;

6) si aucun événement d'achèvement n'est rapporté de l'étape 5), le traitement retourne à l'étape 2).

7.1.14.6 Activation d'un script de numérotation

Un script de numérotation est activé toutes les fois qu'un nouveau descripteur d'événements est appliqué à la terminaison ou qu'un descripteur d'événements incorporé est activé, que ce descripteur d'événements contient un événement d'achèvement de script de numérotation. L'événement d'achèvement de script de numérotation contient un champ eventDM dans le champ des actions demandées. Chaque nouvelle activation d'un script de numérotation commence à l'étape 1 de la procédure précédente avec une chaîne de numérotation actuelle vide. Le contenu éventuel de la chaîne de numérotation actuelle provenant d'une activation précédente est perdu.

Un événement d'achèvement de script de numérotation qui ne contient pas de champ eventDM dans le champ d'actions demandées est considéré comme erroné. Lors de la réception d'un tel événement dans le descripteur EventsDescriptor, une passerelle MG répondra par une réponse d'erreur, notamment l'erreur 457 – Paramètre manquant dans un signal ou dans un événement.

7.1.14.7 Interaction de script de numérotation et traitement d'événement

Pendant que le script de numérotation est activé, la détection est activée pour tous les événements définis dans le paquetage qui contient l'événement d'achèvement de script de numérotation considéré. Le comportement normal des événements (par exemple l'arrêt des signaux à moins que l'événement d'achèvement de numérotation n'ait le fanion KeepActive activé) continue à s'appliquer pour un tel événement détecté éventuel, sauf que:

- les événements du paquetage qui contient l'événement d'achèvement de script de numérotation considéré autre que l'événement d'achèvement lui-même ne sont pas rapportés individuellement et n'ont pas d'effets secondaires à moins d'être activés séparément;
- un événement qui déclenche un événement d'achèvement de correspondance partielle n'est pas reconnu et, par conséquent, n'a pas d'effets secondaires tant qu'il n'est pas retraité à la suite de la reconnaissance d'un événement d'achèvement de script de numérotation. De même, les événements de numérotation mis en mémoire tampon ne sont pas reconnus et n'ont pas réellement d'effet jusqu'à leur traitement.

7.1.14.8 Caractères génériques

Il convient de noter que, si un paquetage contient un événement d'achèvement de script de numérotation, une spécification d'événement constituée du nom du paquetage avec un identificateur ItemID (Property Name) remplacé par un caractère générique activera un script de numérotation; à ces fins, la spécification d'événement doit inclure un champ eventDM conformément au § 7.1.14.6. Si ce paquetage contient également les événements chiffres eux-mêmes, cette forme de spécification d'événement provoque le rapport des événements individuels au contrôleur MGC au fur et à mesure de leur détection.

7.1.14.9 Exemple

A titre d'exemple, considérons le plan de numérotation suivant:

0	Exploitant local
00	Exploitant à grande distance
xxxx	Numéro de poste supplémentaire local (commence par 1 à 7)
8xxxxxx	Numéro local
#xxxxxxx	Prolongement hors site
*xx	Services confort
91xxxxxxxxx	Numéro à grande distance
9011 + jusqu'à 15 chiffres	Numéro international

Si le paquetage de détection DTMF décrit dans le § E.6 est utilisé pour recueillir les chiffres composés, le plan de numérotage illustré précédemment fournit le script de numérotation suivant:

```
(0| 00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxxx|9011x.)
```

7.1.15 Descripteur Statistics (de statistiques)

Le descripteur Statistics donne des informations décrivant l'état et l'usage d'une terminaison pendant son existence dans un contexte spécifique. Le cas échéant, il existe un paquetage de statistiques normalisées pour chaque terminaison (par exemple le nombre d'octets émis et reçus). Les propriétés statistiques particulières qui sont signalées pour une terminaison donnée sont déterminées par les paquetages réalisés par la terminaison. Par défaut, ces statistiques sont signalées lorsque la terminaison est soustraite du contexte par une commande Subtract. Ce comportement peut être annulé en incorporant un descripteur Audit vide dans la commande Subtract. Des statistiques peuvent également être renvoyées par la commande AuditValue ou par toute commande Add/Move/Modify utilisant le descripteur Audit.

Les statistiques sont cumulatives. Le fait de les signaler ne les réinitialise pas. Les statistiques sont réinitialisées lorsqu'une terminaison est soustraite d'un contexte.

7.1.16 Descripteurs de paquetages

Ce descripteur n'est utilisé qu'avec la commande AuditValue. Il renvoie une liste de paquetages réalisés par la terminaison.

7.1.17 Descripteurs ObservedEvents (éléments observés)

Ce descripteur est fourni avec la commande Notify pour informer le contrôleur MGC de l'événement ou des événements détectés. Utilisé avec la commande AuditValue, le descripteur ObservedEvents renvoie les événements contenus dans la mémoire d'événements qui n'ont pas été notifiés. Le descripteur ObservedEvents contient l'identificateur RequestIdentifier du descripteur d'événements qui a déclenché la notification, l'événement ou les événements détectés, en option les temps de détection et tous les paramètres de l'événement observé. Les temps de détection sont consignés avec une précision de centièmes de seconde.

7.1.18 Descripteur Topology

Ce descripteur est utilisé pour spécifier les sens des flux entre les terminaisons d'un contexte. Contrairement aux descripteurs des paragraphes précédents, le descripteur Topology s'applique à un contexte au lieu d'une terminaison. La topologie par défaut d'un contexte est telle que la transmission d'une terminaison soit reçue par toutes les autres terminaisons. L'implémentation du descripteur Topology est facultative. Une passerelle MG qui ne prend pas en charge les descripteurs Topology, mais reçoit une commande qui en contient, renvoie l'erreur 444 (Descripteur non pris en charge ou inconnu), et inclut éventuellement dans le texte du descripteur d'erreur une chaîne contenant le nom du descripteur non pris en charge ("Topology")

Le descripteur Topology apparaît avant les commandes dans un paramètre d'action. Il est possible qu'une action ne contienne qu'un seul descripteur Topology, à la condition que le contexte auquel l'action s'applique existe déjà.

Un descripteur de topologie contient une suite de terminaisons associées de la forme (*T1, T2, association[,StreamId]*). *T1* et *T2* spécifient des terminaisons dans le contexte, éventuellement au moyen du caractère générique ALL ou CHOOSE. Si le champ facultatif StreamId est employé, l'association ne s'applique qu'au flux particulier entre *T1* et *T2* mentionné par StreamId. Si le champ StreamId est omis, la topologie s'applique à tous les flux dans la terminaison. L'élément *association* spécifie comme suit la façon dont les médias s'écoulent entre ces deux terminaisons:

- le triplet (*T1*, *T2*, isolate) signifie que les terminaisons correspondant à *T2* ne reçoivent pas de média des terminaisons correspondant à *T1* et inversement;

- le triplet (*T1, T2,* oneway) signifie que les terminaisons correspondant à *T2* reçoivent des médias des terminaisons correspondant à *T1* mais pas inversement. Dans ce cas, l'utilisation du caractère générique ALL, de façon qu'il y ait des terminaisons correspondant à la fois à *T1* et à *T2*, n'est pas autorisée;
- le triplet (*T1, T2,* bothway) signifie que les terminaisons correspondant à *T2* reçoivent des médias des terminaisons correspondant à *T1* et inversement. Dans ce cas, il est permis d'utiliser des caractères génériques de façon qu'il ait des terminaisons correspondant à la fois à *T1* et à *T2*. Cependant, dans ce dernier cas, aucun bouclage n'est introduit.

Le caractère générique CHOOSE peut également être utilisé dans *T1* et *T2*, aux conditions suivantes:

- l'action (se reporter au paragraphe 8) dont le descripteur de topologie fait partie contient une commande Add dans laquelle un caractère générique CHOOSE est utilisé;
- si un caractère générique CHOOSE apparaît dans *T1* ou *T2*, un nom partiel NE DOIT PAS être spécifié.

Le caractère générique CHOOSE dans un descripteur de topologie correspond à l'identificateur TerminationID que la passerelle MG attribue à la première commande Add qui utilise un caractère générique CHOOSE dans la même action. Une terminaison existante qui correspond à *T1* ou à *T2* dans le contexte auquel une terminaison a été ajoutée est connectée à la terminaison nouvellement ajoutée comme il est spécifié par le descripteur de topologie. Si une terminaison n'est pas mentionnée dans un descripteur de topologie, toute topologie qui lui est associée reste inchangée. Si, toutefois, une nouvelle terminaison est ajoutée dans un contexte, son association avec les autres terminaisons dans le contexte est par défaut bidirectionnelle, à moins qu'un descripteur de topologie ne soit donné dans le but de modifier cela (par exemple, si *T3* est ajouté à un contexte avec *T1* et *T2* et la topologie (*T3, T1* oneway), sa connexion à *T2* sera bidirectionnelle).

Si la topologie est appliquée à un flux particulier (*T1*, *T2*, association, StreamId), la topologie des autres flux entre les terminaisons reste inchangée.

Un descripteur de topologie NE DOIT PAS inclure de combinaison d'associations entre deux terminaisons (Ti, Tj) avec et sans champ facultatif StreamId, afin d'éviter un comportement non défini. Par exemple, (*T1*, *T2*, bothway) et (*T1*, *T2*, isolate, S1) ne doivent pas figurer dans le même descripteur. A la réception d'un tel descripteur de topologie, une passerelle MG répondra par un message d'erreur, notamment l'erreur 421 (Action inconnue ou combinaison d'actions non autorisée). La Figure 7, le tableau et la Figure 8 ci-dessous montrent quelques exemples des conséquences de l'inclusion de descripteurs de topologie dans des actions. Dans ces exemples, il est implicitement admis que les descripteurs de topologie sont appliqués en séquence.

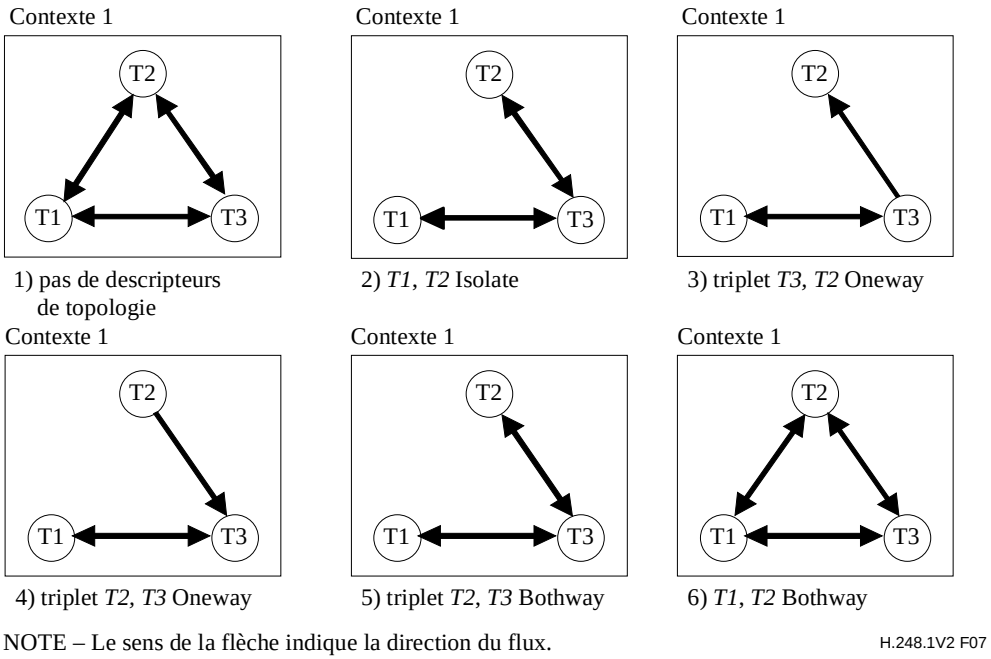
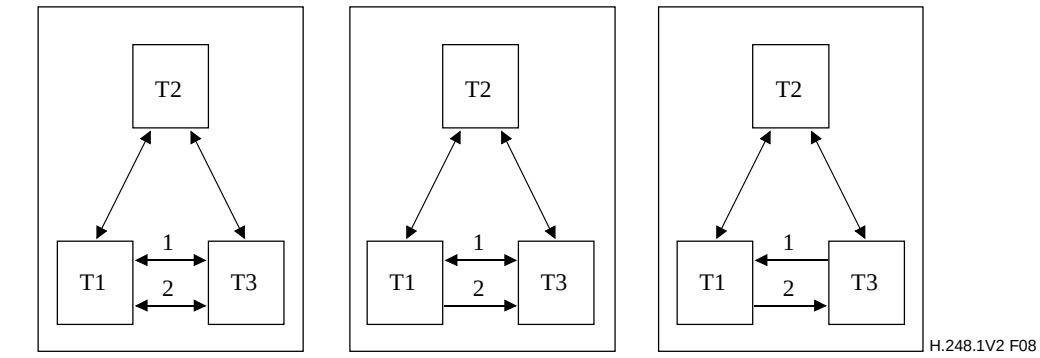


Figure 7/H.248.1 – Exemples de topologie

Topologie	Description
1	Pas de descripteurs de topologie Si aucun descripteur Topology n'est inclus, toutes les terminaisons ont une connexion bilatérale avec toutes les autres terminaisons.
2	*T1*, *T2* Isolate Cette topologie supprime la connexion entre *T1* et *T2*. *T3* possède donc une connexion bilatérale avec *T1* comme avec *T2*. *T1* et *T2* ont une connexion bilatérale avec *T3*.
3	Triplet *T3*, *T2* oneway Une connexion unilatérale de *T3* à *T2* (c'est-à-dire que *T2* reçoit le flux média de *T3*). Une connexion bilatérale existe entre *T1* et *T3*.
4	Triplet *T2*, *T3* oneway Une connexion unilatérale existe de *T2* à *T3*. *T1* et *T3* restent en connexion bilatérale.
5	Triplet *T2*, *T3* bothway *T2* est en connexion bilatérale avec *T3*, ce qui donne le même résultat qu'au chiffre 2 ci-dessus.
6	*T1*, *T2* bothway (*T2*, *T3* bothway et *T1*, *T3* bothway peuvent être implicites ou explicites). Chacune des terminaisons est en connexions bilatérales avec toutes les autres terminaisons.

Une connexion unilatérale doit toujours être implémentée de façon que les autres terminaisons du contexte ne soient pas informées du changement de topologie.



(a) pas de descripteur de topologie (b) *T1,T3,* oneway, 2 (c) *T3,T1,* oneway, 1

Figure 8/H.248.1 – Exemple de topologie au niveau du flux

7.1.19 Descripteur d'erreur

Si un récepteur rencontre une erreur au cours du traitement d'une demande de transaction, il doit inclure un descripteur d'erreur dans sa réponse. Une demande de notification peut aussi contenir un descripteur d'erreur.

Un descripteur d'erreur consiste en un code d'erreur enregistré par l'autorité IANA, éventuellement accompagné d'un texte d'erreur. La Rec. UIT-T H.248.8 contient une liste des codes et des descriptions d'erreur valables.

Un descripteur d'erreur sera spécifié au "niveau le plus profond", approprié sur le plan sémantique à l'erreur décrite et possible, étant donné les problèmes d'analyse de la demande initiale. Un descripteur d'erreur peut se référer à une structure syntaxique ailleurs qu'à l'endroit où il apparaît. Par exemple, le descripteur d'erreur 422 (Erreur de syntaxe dans l'action), pourrait apparaître dans une commande même lorsqu'il se réfère à une structure plus large – l'action – et non à la commande particulière dans laquelle il apparaît.

7.2 Interface de programmation d'application pour les commandes

On trouvera ci-après une interface de programmation d'application (API, *application programming interface*) décrivant les commandes du protocole. Cette interface API vise à illustrer les commandes et leurs paramètres. Elle n'est pas destinée à spécifier une implémentation (par exemple au moyen d'appels de fonction de blocage). Elle décrit les paramètres d'entrée entre parenthèses après le nom de la commande et les valeurs de retour en face de cette commande. Cette interface ne vise que la description. La syntaxe et le codage proprement dits des commandes sont spécifiés dans des paragraphes ultérieurs. L'ordre des paramètres relatifs aux commandes n'est pas fixé. Des descripteurs peuvent servir de paramètres à des commandes dans tout ordre. Les descripteurs DOIVENT être traités dans l'ordre dans lequel ils apparaissent.

Toute réponse à une commande peut contenir un descripteur d'erreur; l'interface API ne montre pas cela explicitement.

Tous les paramètres mis entre crochets ([…]) sont considérés comme étant facultatifs.

7.2.1 Commande Add

La commande Add ajoute une terminaison à un contexte.

Identificateur de terminaison

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Add(TerminationID

[, MediaDescriptor]

[, ModemDescriptor] (*)

[, MuxDescriptor]

[, EventsDescriptor]

[, EventBufferDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor]

[, AuditDescriptor]

)

(*) Le descripteur ModemDescriptor a été désapprouvé dans la Rec. UIT-T H.248.1 version 1 (03/2002).

L'identificateur de terminaison spécifie la terminaison qui doit être ajoutée au contexte. La terminaison est créée ou prise dans le contexte néant. Pour une terminaison existante, l'identificateur de terminaison sera spécifique. Si un caractère générique CHOOSE est employé dans l'identificateur de terminaison, l'identificateur de terminaison choisi sera retourné. Des caractères génériques peuvent être utilisés dans une commande Add, mais un tel usage sera rare. Si le caractère générique concorde avec plusieurs identificateurs de terminaison, toutes les correspondances possibles seront essayées et les résultats seront signalés à chaque fois. L'ordre des essais en cas de multiples correspondances avec des identificateurs de terminaison n'est pas spécifié.

Le paramètre facultatif MediaDescriptor décrit tous les flux médias.

Le paramètre facultatif MuxDescriptor spécifie, le cas échéant, un multiplexeur. Par commodité, si un descripteur de multiplex est présent dans une commande Add et si ce descripteur énumère des terminaisons qui ne sont pas actuellement dans le contexte, ces terminaisons sont ajoutées au contexte comme si des commandes Add individuelles, énumérant ces terminaisons, avaient été invoquées. Si une erreur survient sur une telle commande Add implicite, une erreur 471 (Commande Add implicite pour échec de multiplex) doit être renvoyée et un traitement plus poussé de la commande doit s'arrêter.

Le paramètre EventsDescriptor est facultatif. S'il est présent, il fournit la liste des événements qui devraient être détectés dans la terminaison.

Le paramètre EventBufferDescriptor est facultatif. S'il est présent, il fournit la liste des événements que la passerelle MG est priée de détecter et la mémoire tampon lorsque la commande EventBufferControl est égale à LockStep.

Le paramètre SignalsDescriptor est facultatif. S'il est présent, il fournit la liste des événements qui devraient être appliqués à la terminaison.

Le paramètre DigitMapDescriptor est facultatif. S'il est présent, il donne une définition de script de numérotation qui peut être utilisée dans un descripteur d'événements.

Le paramètre AuditDescriptor est facultatif. S'il est présent, la commande renverra les descripteurs spécifiés dans ce paramètre.

Tous les descripteurs modifiables peuvent être renvoyés par la passerelle MG si un paramètre a été sous-spécifié ou surspécifié. Les descripteurs ObservedEvents, Statistics et Packages ainsi que EventBuffer ne sont retournés que s'ils sont demandés dans le descripteur Audit.

La commande Add NE DOIT PAS être utilisée sur une terminaison avec un état serviceState de "OutofService" (hors service).

7.2.2 Commande Modify

La commande Modify modifie les propriétés d'une terminaison.

Identificateur de terminaison

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
Modify( TerminationID
    [, MediaDescriptor]
    [, ModemDescriptor] (*)
    [, MuxDescriptor]
    [, EventsDescriptor]
    [, EventBufferDescriptor]
    [, SignalsDescriptor]
    [, DigitMapDescriptor]
    [, AuditDescriptor]
)
```

(*) Le descripteur ModemDescriptor a été désapprouvé dans la Rec. UIT-T H.248.1 version 1 (03/2002).

L'identificateur de terminaison peut être spécifique si une seule terminaison doit être modifiée dans le contexte. L'utilisation de caractères génériques dans l'identificateur de terminaison peut convenir pour certaines opérations. Si le caractère générique concorde avec plusieurs identificateurs de terminaison, toutes les correspondances possibles seront essayées et les résultats seront signalés à chaque fois. L'ordre des essais en cas de multiples correspondances avec des identificateurs de terminaison n'est pas spécifié. L'option du caractère CHOOSE est une erreur car la commande Modify ne peut être utilisée qu'avec des terminaisons existantes.

Par souci de commodité, lorsqu'un descripteur de multiplex est présent dans une commande Modify, alors:

- si le nouveau descripteur de multiplex énumère des terminaisons qui ne sont pas effectivement dans le contexte, ces terminaisons doivent y être ajoutées comme si des commandes Add distinctes énumérant les terminaisons avaient été invoquées;
- si des terminaisons énumérées précédemment dans le descripteur de multiplex ne sont plus présentes dans le nouveau descripteur de multiplex, elles seront soustraites du contexte comme si des commandes Substract distinctes énumérant les terminaisons avaient été invoquées.

Les autres paramètres de la commande Modify sont les mêmes que ceux de la commande Add. Les valeurs de retour possibles sont les mêmes que celles de la commande Add.

7.2.3 Commande Subtract

La commande Subtract déconnecte une terminaison de son contexte et renvoie des statistiques sur la participation de cette terminaison dans le contexte.

Identificateur de terminaison

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Subtract(TerminationID

[, AuditDescriptor]

)

(*) Le descripteur ModemDescriptor a été désapprouvé dans la Rec. UIT-T H.248.1 version 1 (03/2002).

L'identificateur de terminaison contenu dans les paramètres d'entrée représente la terminaison qui est en cours de soustraction. L'identificateur de terminaison peut être spécifique ou être un caractère générique indiquant que la totalité (ou un paquetage d'associations) des terminaisons contenues dans le contexte de la commande Subtract doit être soustrait(e). Si le caractère générique concorde avec plusieurs identificateurs de terminaison, toutes les correspondances possibles seront essayées et les résultats seront signalés à chaque fois. L'ordre des essais en cas de multiples correspondances avec des identificateurs de terminaison n'est pas spécifié.

L'utilisation de CHOOSE dans l'identificateur de terminaison est une erreur, car la commande Subtract ne peut être utilisée que sur des terminaisons existantes.

Le caractère ALL peut être utilisé en tant qu'identificateur de contexte ainsi que comme un identificateur de terminaison dans une commande Subtract, ce qui a pour effet de supprimer tous les contextes, de supprimer toutes les terminaisons éphémères et de renvoyer toutes les terminaisons physiques au contexte Null. La soustraction d'une terminaison du contexte Null n'est pas autorisé.

Par souci de commodité, si une terminaison de multiplexage est l'objet d'une commande Substract, alors les terminaisons supports énumérées son descripteur de multiplex sont soustraites du contexte comme si des commandes Substract distinctes énumérant les terminaisons avaient été invoquées.

Par défaut, le paramètre Statistics est renvoyé pour signaler des informations collectées au sujet de la terminaison ou des terminaisons spécifiées dans la commande. Les informations ainsi signalées s'appliquent à l'existence de la ou des terminaisons dans le contexte dont celles-ci sont soustraites.

Le paramètre AuditDescriptor est facultatif. S'il est présent, la commande ne renverra que les descripteurs spécifiés dans ce paramètre, qui peut être vide. S'il est omis, le descripteur Statistics est renvoyé par défaut. Les valeurs de retour possibles sont les mêmes que celles de la commande Add.

Lorsqu'une terminaison fournie est soustraite d'un contexte, ses valeurs de propriété doivent retourner:

- à la valeur par défaut, si elle est spécifiée pour la propriété et n'est pas annulée par la fourniture;
- à la valeur fournie, autrement.

7.2.4 Commande Move

La commande Move déplace une terminaison vers un autre contexte à partir de son contexte actuel, au cours d'une seule opération atomique. La commande Move est la seule commande qui se rapporte à une terminaison dans un contexte différent de celui auquel la commande s'applique. La commande Move ne doit pas être utilisée pour déplacer des terminaisons en direction de ou hors du contexte néant.

Identificateur de terminaison

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
Move( TerminationID
    [, MediaDescriptor]
    [, ModemDescriptor] (*)
    [, MuxDescriptor]
    [, EventsDescriptor]
    [, EventBufferDescriptor]
    [, SignalsDescriptor]
    [, DigitMapDescriptor]
    [, AuditDescriptor]
)
```

(*) Le descripteur ModemDescriptor a été désapprouvé dans la Rec. UIT-T H.248.1 version 1 (03/2002).

L'identificateur de terminaison spécifie la terminaison à déplacer. Il peut être remplacé par un caractère générique mais CHOOSE ne doit pas être utilisé dans l'identificateur de terminaison. Si le caractère générique concorde avec plusieurs identificateurs de terminaison, toutes les correspondances possibles seront essayées et les résultats seront signalés à chaque fois. L'ordre des essais en cas de multiples correspondances avec des identificateurs de terminaison n'est pas spécifié. Le contexte vers lequel la terminaison est déplacée est indiqué par l'identificateur de contexte (ContextId) cible dans Action. Si la dernière terminaison restante est déplacée du contexte, le contexte est supprimé.

La commande Move n'affecte pas les propriétés de la terminaison sur laquelle elle opère, sauf les propriétés explicitement modifiées par les descripteurs inclus dans la commande Move. Le descripteur Audit avec l'option de statistiques, renverra par exemple des statistiques au sujet de la terminaison juste avant le déplacement de celle-ci. Les descripteurs possibles retournés à partir de la commande Move sont les mêmes que pour la commande Add.

Par souci de commodité, si une terminaison de multiplexage est l'objet d'une commande Move, alors les terminaisons supports énumérées son descripteur de multiplex sont aussi déplacées comme si des commandes Move distinctes énumérant les terminaisons avaient été invoquées.

La commande Move NE DOIT PAS être utilisée sur une terminaison avec un état serviceState de "OutofService" (hors service).

7.2.5 Commande AuditValue

La commande AuditValue renvoie les valeurs actuelles des propriétés, des événements, des signaux et des statistiques associés aux terminaisons. Elle peut demander le contenu d'un descripteur ou d'une propriété, d'un événement, d'un signal ou d'une statistique.

Identificateur de terminaison

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

AuditValue(TerminationID,

AuditDescriptor

)

(*) Le descripteur ModemDescriptor a été désapprouvé dans la Rec. UIT-T H.248.1 version 1 (03/2002).

L'identificateur de terminaison peut être spécifique ou générique. Si le caractère générique concorde avec plusieurs identificateurs de terminaison, toutes les correspondances possibles seront essayées et les résultats seront signalés à chaque fois. L'ordre des essais en cas de multiples correspondances avec des identificateurs de terminaison n'est pas spécifié. Si une réponse remplacée par un caractère

générique est demandée, un seul renvoi de commande est généré, et il contient l'union des valeurs de toutes les terminaisons qui correspondent au caractère générique. Cette convention peut réduire le volume de données nécessaires pour auditer un groupe de terminaisons. L'option du caractère CHOOSE est une erreur.

Les descripteurs ou les propriétés, les signaux, les événements et les statistiques distincts peuvent être analysés.

- On peut demander l'analyse d'un descripteur en indiquant dans le descripteur AuditDescriptor l'identité du descripteur souhaité, sans autre information.
- Afin d'analyser une propriété distincte dans le descripteur de média, il faut inclure l'identificateur de flux (facultatif), l'identificateur de groupe (facultatif) et l'identificateur de la propriété pertinents. La valeur courante de la propriété est renvoyée. L'identificateur de groupe est employé dans le cas où le fanion groupe de réserve, destiné à la commande locale, est employé. L'identificateur de groupe 1 correspond au premier groupe (description de session) réservé, l'identificateur de groupe 2 correspond au groupe suivant, etc.
- Afin d'analyser un signal, on fournit l'identificateur de la liste de signaux et/ou l'identificateur du signal pertinents. Les valeurs de tous les paramètres du signal ne sont renvoyées que lorsque le signal est activé, notamment l'indication keepactive, le type de signal, la durée, l'indication d'achèvement du signal et les propriétés définies au niveau du paquetage.
- Afin d'analyser un événement, on fournit l'identificateur de flux (facultatif), l'identificateur de l'événement et l'identificateur de la demande (facultatif) pertinents. Les valeurs de tous les paramètres de l'événement sont renvoyées, notamment: les actions concernant les événements et les paramètres définis au niveau du paquetage.
- Afin d'analyser une statistique, on fournit l'identité de cette statistique. La valeur courante de la statistique est renvoyée. Cette statistique n'est pas réinitialisée.
- Afin d'analyser un paquetage, on fournit l'identité et la version du paquetage. Tous les signaux, propriétés, événements et statistiques définies dans ce paquetage particulier sont renvoyés avec leur valeur courante.

Il est possible d'analyser plusieurs éléments distincts au moyen d'une même demande.

Si une analyse de descripteurs est demandée, les descripteurs appropriés, assortis des valeurs actuelles pour la terminaison, sont renvoyés par la commande AuditValue. Les valeurs apparaissant dans de multiples instances d'un descripteur sont définies de façon à être des variantes prises en charge, chaque paramètre d'un descripteur étant considéré comme indépendant.

Le descripteur ObservedEvents renvoie une liste d'événements contenus dans la mémoire d'événements. Si le descripteur ObservedEvents est audité pendant qu'un script de numérotation est actif, le descripteur ObservedEvents renvoyé comprend également un événement d'achèvement de script de numérotation qui indique la chaîne de numérotation courante sans donner de méthode de terminaison.

La mémoire EventBuffer renvoie le paquetage des événements et les valeurs de paramètre associées actuellement activées dans le descripteur EventBuffer. Le descripteur Packages renvoie une liste de paquetages réalisés par la terminaison. Le descripteur DigitMap renvoie le nom ou la valeur du script de numérotation actuel pour la terminaison. Le descripteur DigitMap demandé dans une commande AuditValue avec l'identificateur de terminaison ALL renvoie tous les scripts de numérotation dans la passerelle. Le descripteur Statistics renvoie les valeurs actuelles de toutes les statistiques enregistrées au sujet de la terminaison. La spécification d'un descripteur Audit vide se traduit par le renvoi du seul identificateur de terminaison. Cela peut être utile afin d'obtenir une liste d'identificateurs de terminaison, en tant que caractère générique de remplacement. Les Annexes A et B contiennent une syntaxe spéciale permettant de présenter une telle liste sous forme condensée,

de sorte que l'étiquette de commande AuditValue n'a pas besoin d'être répétée pour chaque identificateur de terminaison.

Les résultats (à savoir "spécifique", "néant" ou "générique") de la commande AuditValue dépendent du contexte. (Il est à noter que l'identificateur de contexte "Tous" n'inclut pas le contexte néant.) L'identificateur de terminaison peut être spécifique ou générique.

Les exemples suivants montrent les renvois qui sont effectués lorsque le contexte ou la terminaison est générique et qu'une réponse générique a été spécifiée.

Supposons que la passerelle a 4 terminaisons: t1/1, t1/2, t2/1 and t2/2. Supposons que les terminaisons t1/* ont implémenté les paquetages aaa et bbb tandis que les terminaisons t2/* ont implémenté les paquetages ccc et ddd. Supposons que le contexte 1 contient t1/1 et t2/1 et que le contexte 2 contient t1/2 et t2/2.

Commande:

Context=1{AuditValue=t1/1{Audit{Packages}}}

Renvoi:

Context=1{AuditValue=t1/1{Packages{aaa,bbb}}}

Commande:

Context=*{AuditValue=t2/*{Audit{Packages}}}

Renvoi:

Context=1{AuditValue=t2/1{Packages{ccc,ddd}}},
Context=2{AuditValue=t2/2{Packages{ccc,ddd}}}

Commande:

Context=*{W-AuditValue=t1/*{Audit{Packages}}}

Renvoi:

Context=*{W-AuditValue=t1/*{Packages{aaa,bbb}}}

NOTE – Une réponse générique peut aussi être employée pour d'autres commandes telles que Substract.

Le tableau ci-dessous décrit d'autres informations qui peuvent être obtenues par la commande AuditValue:

Identificateur de contexte	Identificateur de terminaison	Informations obtenues
Spécifique	Caractère générique	Analyse des terminaisons concordantes dans un contexte
Spécifique	Spécifique	Analyse d'une terminaison unique dans un contexte
Néant	Racine	Analyse de l'état et des événements d'une passerelle MG
Néant	Caractère générique	Analyse de toutes les terminaisons concordantes contenues dans le contexte néant
Néant	Spécifique	Analyse d'une terminaison unique hors de tout contexte
Tous	Caractère générique	Analyse de toutes les terminaisons concordantes et des contextes auxquels elles sont associées
Tous	Racine	Liste de tous les identificateurs de contexte (cette liste devrait être renvoyée au moyen de réponses d'action multiples, chacune contenant un identificateur de contexte de la liste)
Tous	Spécifique	Identificateur de contexte (non nul) dans lequel la terminaison existe effectivement

7.2.6 AuditCapabilities

La commande AuditCapabilities renvoie les valeurs possibles de propriétés, d'événements, de signaux et de statistiques associés à des terminaisons. Elle peut être demandée pour les contenus d'un descripteur ou d'une propriété, d'un événement, d'un signal ou d'une statistique

Identificateur de terminaison

[,MediaDescriptor]

[,ModemDescriptor](*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

AuditCapabilities(TerminationID,

AuditDescriptor)

(*) Le descripteur Modem Descriptor a été désapprouvé dans la Rec. UIT-T H.248.1 version 1 (03/2002).

Les descripteurs ou les propriétés, les signaux, les événements et les statistiques distincts peuvent être analysés.

- On peut demander l'analyse d'un descripteur entier en indiquant dans le descripteur AuditDescriptor le descripteur souhaité, sans autre information.
- Afin d'analyser une propriété distincte dans le descripteur de média, il faut inclure l'identificateur de flux (facultatif) et l'identificateur de la propriété. Une liste des valeurs possibles de la propriété est renvoyée.

- Afin d'analyser un signal, on fournit l'identificateur de la liste de signaux et/ou l'identificateur du signal pertinent. Une liste des valeurs possibles associées à chacun des paramètres du signal est renvoyée (notamment l'indication "keepactive", le type de signal, la durée, l'indication d'achèvement du signal et les propriétés définies au niveau du paquetage).
- Afin d'analyser un événement, on fournit l'identificateur de flux (facultatif), l'identificateur de l'événement et l'identificateur de la demande (facultatif) pertinents. Une liste des valeurs possibles associées à chacun des paramètres de l'événement est renvoyée (notamment: les actions concernant les événements et les paramètres définis au niveau du paquetage).
- Afin d'analyser une statistique, on fournit l'identité de cette statistique. Les valeurs possibles de la statistique sont renvoyées. Cette statistique n'est pas réinitialisée.

Si une analyse de descripteurs est demandée, les descripteurs appropriés, assortis des valeurs possibles pour la terminaison, sont renvoyés par la commande AuditCapabilities. Certains descripteurs peuvent être répétés lorsqu'il existe plusieurs valeurs possibles.

Si une réponse remplacée par un caractère générique est demandée, un seul renvoi de commande est généré, et il contient l'union des valeurs de toutes les terminaisons qui correspondent au caractère générique. Cette convention peut réduire le volume de données nécessaires pour auditer un groupe de terminaisons.

Si une propriété, un signal, un événement ou une statistique sont analysés, les propriétés appropriées, les événements concernant les signaux et les statistiques avec les capacités de la terminaison sont renvoyées par la commande AuditCapabilities.

L'interprétation quant aux capacités qui sont requises pour diverses valeurs d'identificateur de contexte et de terminaison est la même que pour la commande AuditValue.

Le paramètre EventsDescriptor renvoie la liste des événements possibles concernant la terminaison ainsi que la liste de toutes les valeurs possibles pour le paramètre EventsDescriptor. Le paramètre EventBufferDescriptor renvoie les mêmes informations que le paramètre EventsDescriptor, ainsi que la liste de toutes les valeurs possibles pour le paramètre EventsDescriptor. Le paramètre SignalsDescriptor renvoie la liste des signaux dont l'application à la terminaison est possible, ainsi que la liste de toutes les valeurs possibles des paramètres de signaux. Le paramètre StatisticsDescriptor renvoie le nom des statistiques enregistrées pour la terminaison. Le paramètre ObservedEventsDescriptor renvoie les noms des événements actifs dans la terminaison. Les descripteurs DigitMap et Packages ne sont pas autorisés dans la commande AuditCapability.

Le tableau suivant montre d'autres informations qui peuvent être obtenues au moyen de la commande AuditCapabilities:

Identificateur de contexte	Identificateur de terminaison	Information obtenue
Spécifique	Caractère générique	Analyse des terminaisons concordantes dans un contexte
Spécifique	Spécifique	Analyse d'une terminaison unique dans un contexte
Néant	Racine	Analyse de l'état et des événements dans une passerelle MG
Néant	Caractère générique	Analyse de toutes les terminaisons concordantes contenues dans le contexte néant
Néant	Spécifique	Analyse d'une terminaison unique hors de tout contexte
Tous	Caractère générique	Analyse de toutes les terminaisons concordantes et des contextes auxquelles elles sont associées
Tous	Racine	Voir la commande AuditValue
Tous	Spécifique	Voir la commande AuditValue

7.2.7 Commande Notify

La commande Notify permet à la passerelle MG de signaler au contrôleur MGC que des événements se produisent dans cette passerelle.

Identificateur de terminaison

Notify(TerminationID,

ObservedEventsDescriptor,

[ErrorDescriptor])

Le paramètre TerminationID spécifie la terminaison qui émet la commande Notify. L'identificateur de terminaison doit être un nom entièrement qualifié.

Le paramètre ObservedEventsDescriptor contient l'identification de demande et une liste d'événements que la passerelle MG a détectés, dans l'ordre de leur détection. Chaque événement de la liste est assorti des paramètres qui lui sont associés ainsi qu'éventuellement de l'indication du moment de cette détection. Les procédures pour envoyer des commandes Notify avec l'identificateur RequestID égal à 0 sont l'objet d'une étude plus poussée.

Les commandes Notify avec un identificateur RequestID différent de 0 ne doivent se produire qu'en tant que résultat de la détection d'un événement spécifié par un descripteur Events actif sur la terminaison concernée.

L'identificateur de demande renvoie le paramètre RequestID du descripteur d'événements qui a activé la commande Notify. Il sert à associer la notification à la demande qui l'a activée. Les événements contenus dans la liste doivent avoir été demandés par l'intermédiaire du descripteur Events de déclenchement ou par le biais du descripteur Events incorporé, à moins que l'identificateur RequestID ne soit 0 (ce qui est l'objet d'une étude plus poussée).

Le descripteur ErrorDescriptor peut être envoyé dans la Commande Notify à la suite de l'erreur 518 (Mémoire tampon d'événements pleine).

7.2.8 ServiceChange

La commande ServiceChange permet à la passerelle MG de signaler au contrôleur MGC qu'une terminaison ou un groupe de terminaison est sur le point d'être mis hors service ou vient d'être remis en service. Le contrôleur MGC peut indiquer que la ou les terminaisons doivent être mises hors

service ou remises en service. La passerelle MG peut signaler au contrôleur MGC que la capacité d'une terminaison a changé. Cette commande permet également à un contrôleur MGC de transférer le contrôle d'une passerelle MG à un autre contrôleur MGC.

TerminationID,

[ServiceChangeDescriptor]

ServiceChange(TerminationID,

ServiceChangeDescriptor

)

Le paramètre TerminationID spécifie la ou les terminaisons qui sont mises hors service ou remises en service. Le remplacement de noms de terminaison par des caractères génériques est autorisé, sauf que le caractère CHOOSE ne doit pas être utilisé. L'utilisation de l'identificateur de terminaison "racine" indique qu'un changement de service affecte le paquetage de la passerelle MG.

Le descripteur de changement de service (ServiceChangeDescriptor) contient les paramètres suivants, selon les besoins:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeDelay;
- ServiceChangeAddress;
- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMgcId;
- TimeStamp;
- ServiceChangeInfo.

Le paramètre ServiceChangeMethod spécifie le type de changement de service qui va se produire ou qui s'est produit:

1) la valeur "Graceful": indique que les terminaisons spécifiées seront mises hors service après le délai spécifié dans le paramètre ServiceChangeDelay; les connexions établies ne sont pas encore affectées mais le contrôleur MGC doit s'abstenir d'établir de nouvelles connexions et doit tenter de supprimer progressivement les connexions existantes sur la ou les terminaisons affectées par la commande serviceChange. Il convient que la passerelle règle à "hors service" le paramètre serviceState au moment de l'expiration du délai ServiceChangeDelay ou au moment du retrait de la terminaison d'un contexte actif (selon le moment qui survient le premier);

2) la valeur "Forced": indique que les terminaisons spécifiées ont été mises brusquement hors service et que toutes connexions établies en association avec ces terminaisons peuvent être perdues. Pour les terminaisons qui ne sont pas de type racine, le contrôleur MGC est chargé de nettoyer le contexte (le cas échéant) auquel la terminaison défaillante est associée. Au minimum, la terminaison doit être soustraite du contexte. Il convient que l'état serviceState de la terminaison soit "out of service" (hors service). Pour une terminaison de type racine, le contrôleur MGC peut supposer que toutes les connexions sont perdues au niveau de la passerelle MG et peut don considérer que toutes les terminaisons ont été soustraites;

3) la valeur "Restart": indique que le service sera rétabli sur les terminaisons spécifiées après l'expiration du délai ServiceChangeDelay. Il convient que l'état serviceState soit réglé à "inService" (en service) au moment de l'expiration du délai ServiceChangeDelay;

4) la valeur "Disconnected": toujours appliquée à l'identificateur de terminaison racine, indique que la passerelle MG a perdu la communication avec le contrôleur MGC mais que cette communication a été rétablie par la suite (éventuellement après qu'il ait essayé auprès d'autres contrôleurs MGC énumérés sur une liste établie au préalable). Comme l'état de la passerelle MG peut avoir changé, le contrôleur MGC peut souhaiter utiliser la commande Audit pour resynchroniser son état avec celui de la passerelle MG;

5) la valeur "Handoff": envoyée par le contrôleur MGC à la passerelle MG, indique que le contrôleur MGC va être mis hors service et qu'une nouvelle association de contrôleur MGC doit être établie. Envoyée de la passerelle MG au contrôleur MGC, cette valeur indique que la passerelle MG essaye d'établir une nouvelle association en conformité avec la valeur "Handoff" reçue du contrôleur MGC auquel elle était précédemment associée;

6) la valeur "Failover": envoyée de la passerelle MG au contrôleur MGC, indique que la passerelle MG primaire est hors service et qu'une passerelle MG secondaire prend le relais. La méthode serviceChange est aussi envoyée de la passerelle MG au contrôleur MGC lorsque la passerelle MG détecte que celui-ci a échoué;

7) toute autre valeur dont la signification est mutuellement comprise entre la passerelle MG et le contrôleur MGC.

Le paramètre ServiceChangeReason spécifie la raison pour laquelle le changement de service s'est produit ou va se produire. Il se compose d'un jeton alphanumérique (enregistré par l'autorité IANA) et, facultativement, d'une chaîne explicative.

Le paramètre ServiceChangeAddress spécifie l'adresse (par exemple, le numéro d'accès IP pour réseaux en protocole IP) à utiliser pour les communications subséquentes. Il peut être spécifié dans le descripteur de paramètre d'entrée ou dans le descripteur de résultat renvoyé. Les paramètres ServiceChangeAddress et ServiceChangeMgcId ne doivent pas être présents tous les deux dans le descripteur ServiceChange ou le descripteur ServiceChangeResult. Le paramètre ServiceChangeAddress fournit une adresse à utiliser à l'intérieur du contexte d'une association en cours de négociation, alors que le paramètre ServiceChangeMgcId fournit une adresse de remplacement où il convient que la passerelle MG cherche à établir une autre association. Il convient de noter que l'emploi du paramètre ServiceChangeAddress n'est pas encouragé. Les contrôleurs MGC et les passerelles MG doivent être en mesure de prendre en charge le paramètre ServiceChangeAddress, qu'il soit une adresse complète ou simplement un numéro de port dans le cas des transports TCP.

Le paramètre facultatif ServiceChangeDelay est exprimé en secondes. Si le délai est absent ou mis à zéro, il convient que sa valeur soit considérée comme étant néant. Dans le cas d'une méthode de changement de service progressive ("graceful"), un délai néant indique que le contrôleur MGC doit attendre l'extinction naturelle des connexions existantes et qu'il ne doit pas en établir de nouvelles. Pour la valeur "graceful" uniquement, un délai néant signifie que la passerelle MG ne doit pas régler l'état serviceState à "out of service" (hors service) jusqu'à ce que la terminaison soit dans le contexte néant.

Le paramètre facultatif ServiceChangeProfile spécifie le profil (éventuel) du protocole pris en charge. Ce paramètre indique la version du profil pris en charge.

Le paramètre facultatif ServiceChangeVersion contient la version du protocole et il est utilisé si la négociation de la version du protocole se produit (se reporter au § 11.3).

Le paramètre facultatif TimeStamp (horodatage) spécifie l'heure effective telle qu'elle est conservée par l'expéditeur. En tant que telle, ce n'est pas nécessairement l'heure absolue conforme, par exemple, à un faisceau horaire local: c'est plutôt un temps de début arbitraire par rapport auquel les horodateurs ultérieurs expédiés par un émetteur au cours de cette association seront comparés. Il peut être utilisé par le récepteur afin de déterminer comment sa notion du temps diffère de celle de

son correspondant. Le paramètre TimeStamp est envoyé avec une précision de centièmes de seconde.

Le paramètre facultatif Extension peut contenir n'importe quelle valeur dont la signification est mutuellement comprise par la passerelle MG et le contrôleur MGC. Le paramètre facultatif ServiceChangeInfo peut contenir le paquetage/la propriété/le signal/l'événement/la statistique associés au motif de la modification de service.

Une commande ServiceChange spécifiant la racine ("Root") pour l'identificateur de terminaison et une méthode ServiceChangeMethod égale à "Restart" est une commande d'enregistrement permettant à une passerelle MG d'annoncer son existence au contrôleur MGC. La passerelle MG peut procéder à l'enregistrement en spécifiant la racine ("Root") pour l'identificateur de terminaison et une méthode ServiceChangeMethod égale à "Failover" lorsqu'elle détecte des défaillances de la part des contrôleurs MGC. La passerelle MG est censée être profilée avec le nom d'un contrôleur MGC primaire et, facultativement, d'un certain nombre de contrôleurs MGC de remplacement. L'acquittement de la commande ServiceChange met fin au processus d'enregistrement, sauf si le contrôleur MGC a renvoyé un paramètre ServiceChangeMgcId de rechange tel que décrit dans l'alinéa suivant. La passerelle MG peut spécifier, dans le paramètre ServiceChangeAddress contenu dans le paramètre ServiceChangeDescriptor, l'adresse de transport de changement de service que le contrôleur MGC utilisera pour envoyer des messages. La passerelle MG peut spécifier une adresse dans le paramètre ServiceChangeAddress contenu dans la demande de changement de service et le contrôleur MGC peut faire de même dans la réponse à la demande de changement de service. Dans un cas comme dans l'autre, le destinataire doit utiliser l'adresse fournie comme la destination de toutes les demandes subséquentes de transaction contenues à l'intérieur de l'association. En même temps, comme il est indiqué dans le paragraphe 9, les réponses de transactions et les indications en attente doivent être envoyées à l'adresse d'origine des demandes correspondantes. Il doit en être ainsi même si cela implique une messagerie supplémentaire car les commandes et les réponses ne peuvent pas être mises en paquets paquetage. Le paramètre d'horodatage TimeStamp doit être envoyé avec une commande d'enregistrement et sa réponse.

Le contrôleur MGC peut renvoyer un paramètre ServiceChangeMgcId décrivant le contrôleur MGC que la passerelle MG doit de préférence approcher pour la suite du service. Dans ce cas, la passerelle MG doit réémettre la commande ServiceChange vers le nouveau contrôleur MGC. Le contrôleur MGC spécifié dans un éventuel paramètre ServiceChangeMgcId doit être approchée avant tout autre contrôleur MGC de remplacement. Selon un message de transfert envoyé par le contrôleur MGC à la passerelle MG, le paramètre ServiceChangeMgcId indique le nouveau contrôleur MGC qui prendra la suite du contrôleur MGC actuel.

Le retour provenant de ServiceChange est vide sauf si l'identificateur de terminaison "racine" est utilisé. Dans ce cas, il contient les paramètres suivants, selon les besoins:

- ServiceChangeAddress, si le contrôleur MGC récepteur souhaite spécifier une nouvelle destination pour les messages provenant de la passerelle MG pour le reste de l'association;
- ServiceChangeMgcId, si le contrôleur MGC récepteur ne souhaite pas entretenir une association avec la passerelle MG;
- ServiceChangeProfile, si le récepteur souhaite négocier le profil à utiliser pour l'association. Le profil (nom et version) n'est renvoyé en réponse que dans le cas où le contrôleur MGC n'est pas en mesure de prendre en charge les profils spécifiés dans la demande ServiceChangeRequest. La réponse renvoyée doit indiquer le profil et la version pris en charge ou mentionner "NoProfile", si aucun profil n'est pris en charge. A la réception d'un profil dans la réponse, la passerelle MG peut maintenir sa relation avec le contrôleur MGC actuel ou prendre contact avec les contrôleurs secondaires et établir une relation avec eux. Si le profil n'est pas renvoyé, le contrôleur MGC emploiera les capacités indiquées par le profil qui est indiqué dans la demande de modification de service;

- ServiceChangeVersion, si le récepteur souhaite négocier la version du protocole à utiliser pour l'association.

Les raisons de changement de service (ServiceChangeReasons) suivantes sont définies. Cette liste peut être étendue par un enregistrement IANA comme il est souligné dans le § 14.3.

```
900   Rétablissement du service
901   Démarrage à froid
902   Démarrage à chaud
903   Modification commandée par le contrôleur MGC
904   Mauvais fonctionnement de terminaison
905   Mise hors service de terminaison
906   Perte de connexité dans les couches inférieures (par exemple
      synchronisation en aval)
907   Défaillance de transmission
908   Défaillance de passerelle MG imminente
909   Défaillance de contrôleur MGC imminente
910   Défaillance de capacité média
911   Défaillance de capacité modem
912   Défaillance de capacité multiplex
913   Défaillance de capacité signal
914   Défaillance de capacité événement
915   Perte d'état
916   Modification des paquetages
917   Modification de la capacité
```

7.2.9 Manipulation et audit d'attributs de contexte

Les commandes du protocole exposées dans les paragraphes précédents s'appliquent aux terminaisons. Le présent paragraphe spécifie la façon de manipuler et auditer des contextes.

Une action peut contenir des instructions de manipulation et d'audit de contexte (voir le paragraphe 8).

Une demande d'action envoyée à une passerelle MG peut contenir une demande d'audit d'attributs d'un contexte.

Le contrôleur MGC peut analyser un contexte particulier afin de déterminer la valeur actuelle de ses propriétés distinctes. Il peut déterminer les valeurs actuelles de tous les contextes existants (non NULL) en attribuant à l'identificateur de contexte, dans la demande d'analyse, la valeur ALL. Si des attributs de contexte sont ajoutés ou ont été modifiés en même temps que la demande d'analyse, la ou les valeurs seront renvoyées après que l'action a été effectuée.

Le tableau ci-après illustre les informations pouvant être obtenues dans un contexte d'audit:

Identificateur de contexte	Identificateur de terminaison	Audit
Spécifique	Sans objet	Valeur de l'attribut d'un contexte dans le contexte spécifié
Néant	Sans objet	Non autorisé.
Tous	Sans objet	Valeurs actuelles pour les contextes existants (non NULL), l'identificateur de contexte ayant, dans la demande d'analyse, la valeur ALL. Une réponse pour l'identificateur de contexte ALL est présentée au moyen d'une réponse par contexte relative à une action.

Une action peut également contenir une demande de modification des attributs d'un contexte.

Les propriétés d'un contexte qui peuvent être contenues dans une réponse d'action sont utilisées pour renvoyer des informations au contrôleur MGC. Il peut s'agir d'informations demandées par un audit d'attributs de contexte ou de détails relatifs à l'effet de la manipulation d'un contexte.

Si une passerelle MG reçoit une action contenant à la fois une demande d'audit d'attributs de contexte et une demande de manipulation de ces attributs, la réponse DOIT contenir les valeurs des attributs après le traitement de la demande de manipulation.

7.2.10 Syntaxe de commande générique

Le protocole peut être codé soit en format binaire soit en format alphanumérique. Il y a lieu que les contrôleurs MGC prennent en charge ces deux formats de codage. Les passerelles MG peuvent prendre en charge les deux formats.

La syntaxe de protocole pour le format binaire du protocole est définie dans l'Annexe A. L'Annexe C spécifie le codage des descripteurs Local et Remote en vue d'une utilisation avec le format binaire.

L'Annexe B donne en formalisme BNF augmenté (ABNF, *augmented Backus-Naur form*) le codage alphanumérique du protocole selon RFC 2234. Le protocole SDP est utilisé pour le codage des descripteurs Local et Remote en vue d'une utilisation avec le format alphanumérique tel que modifié au § 7.1.8.

8 Transactions

Les commandes acheminées entre le contrôleur MGC et la passerelle MG sont groupées en transactions, dont chacune est désignée par un identificateur de transaction. Des transactions se composent d'une ou de plusieurs actions. Une action se compose d'une série non vide de commandes, de modifications de la propriété de contexte ou d'analyse de la propriété de contexte dont le fonctionnement est limité à un seul contexte. Par conséquent, chaque action spécifie normalement un identificateur de contexte. Il y a cependant deux circonstances dans lesquelles un identificateur de contexte spécifique n'est pas associé à une action. L'une est le cas d'une modification d'une terminaison extérieure à un contexte. L'autre concerne l'emplacement de création d'un nouveau contexte demandé par le contrôleur à la passerelle. Les relations entre transactions, actions et commandes sont décrites à la Figure 9.

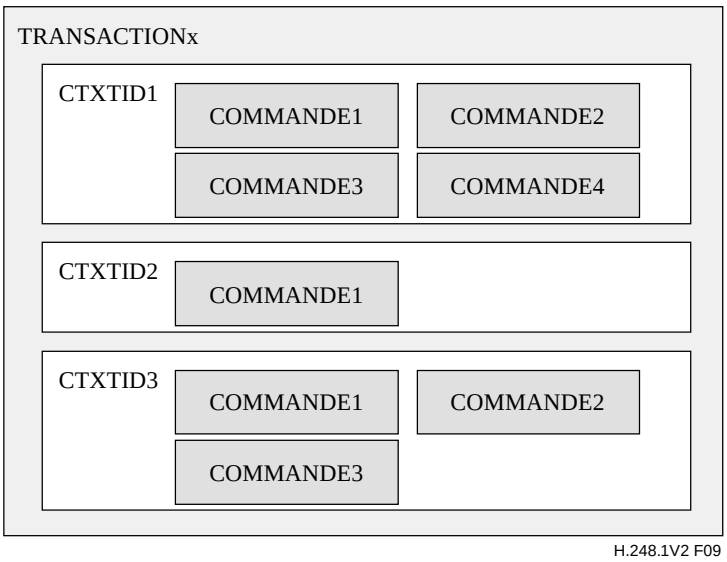


Figure 9/H.248.1 – Transactions, actions et commandes

Les transactions sont présentées sous la forme de demandes de transaction (TransactionRequest). Les réponses correspondantes à une demande de transaction sont reçues dans une seule réponse, éventuellement précédée d'un nombre de messages portant sur les transactions en cours (se reporter au § 8.2.3).

Les transactions garantissent un traitement ordonné des commandes, c'est-à-dire une exécution séquentielle de celles-ci dans le cadre d'une transaction. L'ordre des transactions n'est pas garanti car les transactions peuvent être exécutées dans un ordre quelconque ou simultanément.

Dès la première commande défaillante dans une transaction, le traitement des commandes restant à exécuter dans cette transaction est interrompu. Si une commande contient un identificateur de terminaison remplacé par un caractère générique, cette commande est exécutée avec chacun des identificateurs de terminaison réels qui correspondent à ce caractère. Une réponse dans TransactionReply est incluse pour chaque identificateur de terminaison même si une ou plusieurs instances ont donné lieu à une erreur. Si un identificateur de terminaison correspondant à un caractère générique donne lieu à une erreur lors de son exécution, aucune commande n'est exécutée après celle qui a été remplacée par ce caractère générique.

Les commandes peuvent être marquées comme étant "facultatives", ce qui peut neutraliser ce comportement, c'est-à-dire que, si une commande marquée comme facultative donne lieu à une erreur, les commandes subséquentes de la transaction seront exécutées. Si une commande échoue, la passerelle MG rétablira dans la mesure du possible l'état qui existait avant la tentative d'exécution de la commande avant de poursuivre le traitement des commandes.

Une réponse de transaction effectuée (TransactionReply) contient les résultats pour toutes les commandes contenues dans la demande de transaction correspondante. Cette réponse contient les valeurs de retour pour les commandes qui ont été exécutées normalement ainsi que l'identificateur et le descripteur d'erreur de toute commande ayant échoué. Une réponse de transaction en cours (TransactionPending) est utilisée pour signaler périodiquement au récepteur qu'une transaction n'a pas encore été effectuée mais est en cours de traitement actif.

Il CONVIENT que les applications implémentent lors de chaque transaction un temporisateur de niveau d'application dont l'expiration devrait entraîner la réémission de la demande. La réception d'une réponse de transaction effectuée devrait désarmer le temporisateur. La réception d'une réponse de transaction en cours devrait réarmer le temporisateur.

8.1 Paramètres communs

8.1.1 Identificateurs de transaction

Les transactions sont identifiées par un identificateur de transaction qui est attribué par l'émetteur et qui est unique dans le domaine de celui-ci. Une réponse contenant un descripteur d'erreur visant à signaler que l'identificateur de transaction est absent d'une demande doit utiliser l'identificateur de transaction 0 dans la réponse de transaction correspondante.

8.1.2 Identificateurs de contexte

Les contextes sont identifiés par des identificateurs qui sont attribués par la passerelle MG et qui sont uniques dans le domaine de celle-ci. Le contrôleur MGC doit utiliser l'identificateur de contexte fourni par la passerelle MG dans toutes les transactions subséquentes qui se rapportent à ce contexte. Le protocole fait référence à une valeur distinctive que le contrôleur MGC peut utiliser pour se référer à une terminaison qui n'est pas actuellement associée à un contexte, c'est-à-dire l'identificateur de contexte *néant*.

Le caractère générique CHOOSE sert à demander à la passerelle média de créer un nouveau contexte.

Le contrôleur MGC peut utiliser le caractère générique ALL pour adresser tous les contextes présents sur la passerelle MG. Le contexte néant n'est pas inclus lorsque le caractère générique ALL est utilisé.

Le contrôleur MGC n'emploiera pas les identificateurs de contexte partiellement spécifiés contenant les caractères génériques CHOOSE ou ALL.

8.2 Interface de programmation d'application pour les transactions

L'interface de programmation d'application (API, *application programming interface*) ci-dessous décrit les transactions du protocole. Cette interface API vise à illustrer les transactions et leurs paramètres. Elle n'est pas destinée à spécifier une implémentation (par exemple au moyen d'appels de fonction de blocage). Elle décrira les paramètres d'entrée et les valeurs de retour censés devoir être utilisés par les diverses transactions du protocole à partir d'un niveau très élevé. La syntaxe et les codages des transactions seront spécifiés dans des paragraphes ultérieurs.

8.2.1 Demande de transaction (TransactionRequest)

La demande de transaction est invoquée par l'émetteur. Il y a une seule transaction par invocation de demande. Une demande contient une ou plusieurs actions, dont chacune spécifie son contexte cible, ainsi qu'une ou plusieurs commandes par contexte.

TransactionRequest(TransactionId {

ContextID {Command … Command},

. . .

ContextID {Command … Command } })

Le paramètre TransactionID doit toujours spécifier une valeur en vue d'une corrélation ultérieure avec la réponse de type transaction effectuée ou transaction en cours en provenance du récepteur.

Le paramètre ContextID doit toujours spécifier une valeur à rattacher à toutes les commandes qui font suite soit à la prochaine spécification d'un paramètre ContextID soit à la fin de la demande de transaction, selon le message qui arrive en premier.

Le paramètre Command désigne l'une des commandes mentionnées au § 7.2 (Interface de programmation d'application pour les commandes).

8.2.2 Réponse de Transaction

La réponse de transaction effectuée est invoquée par le récepteur. Il y a une seule invocation de réponse par transaction. Une réponse contient une ou plusieurs actions, dont chacune spécifie son contexte cible, ainsi qu'une ou plusieurs réponses par contexte. La réponse de transaction est invoquée par le récepteur lorsqu'il a traité la demande de transaction.

Une demande de transaction est traitée lorsque:

- toutes les actions dans la demande de transaction ont été exécutées;
- une erreur est produite au cours du traitement de la demande de transaction, ailleurs que dans une commande optionnelle.

Une commande est traitée lorsque tous les descripteurs dans cette commande ont été traités.

Un descripteur Signals est considéré comme ayant été traité lorsqu'il a été établi que sa syntaxe est valable, que les signaux demandés sont pris en charge et qu'ils ont été placés en file d'attente dans le but d'être appliqués.

Un descripteur Events ou EventBuffer est considéré comme ayant été traité lorsqu'il à été établi que sa syntaxe est valable, que les événements demandés peuvent être observés que tout signal intégré peut être émis, que tout événement intégré peut être détecté et que la passerelle MG a été placée dans un état permettant la détection des événements.

TransactionReply(TransactionID {

ContextID { Response … Response },

. . .

ContextID { Response … Response } })

Le paramètre TransactionID doit être le même que celui de la demande TransactionRequest correspondante.

Le paramètre ContextID doit toujours spécifier une valeur à rattacher à toutes les réponses pour l'action. Cet identificateur peut être spécifique, tous ou néant.

Chacun des paramètres de réponse représente une valeur de retour comme indiqué au § 7.2 ou un descripteur d'erreur si l'exécution de la commande a rencontré une erreur. Les commandes postérieures au point de défaillance ne sont pas traitées et des réponses ne sont donc pas émises pour elles.

Une exception à cette règle existe si une commande a été marquée comme étant facultative dans la demande de transaction. Si la commande facultative produit une erreur, la transaction continue à s'exécuter de sorte que des réponses seront émises après une erreur.

Le paragraphe 7.1.19 intitulé (Descripteur d'erreur) décrit la production des descripteurs d'erreur. Le texte ci-après traite de plusieurs cas individuels.

Si le récepteur rencontre une erreur lors du traitement d'un identificateur de contexte, la réponse d'action demandée sera composée de l'identificateur de contexte et d'un seul descripteur d'erreur, 422 (Erreur de syntaxe dans la commande Action).

Si le récepteur rencontre une erreur telle qu'il ne puisse pas déterminer une action autorisée, il renvoie une réponse de transaction effectuée composée de l'identificateur de transaction et d'un seul descripteur d'erreur, 422 (Erreur de syntaxe dans la commande Action). Si la fin d'une action ne peut pas être déterminée de manière fiable mais qu'il est possible d'analyser une ou plusieurs commandes, le récepteur les traite et envoie ensuite 422 (Erreur de syntaxe dans la commande Action) en tant que dernière action pour la transaction. Si le récepteur rencontre une erreur telle qu'il ne puisse pas déterminer une transaction autorisée, il renvoie une réponse de transaction

effectuée composée de l'identificateur de transaction néant et d'un seul descripteur d'erreur 403 (Erreur de syntaxe dans la demande de transaction).

Si la fin d'une transaction ne peut pas être déterminée de manière fiable mais qu'il est possible d'analyser une ou plusieurs actions, le récepteur les traite et envoie ensuite 403 (Erreur de syntaxe dans la transaction) en tant que dernière réponse d'action pour la transaction. Si aucune Action ne peut être analysée, il renvoie 403 (Erreur de syntaxe dans la transaction) en tant que seule réponse.

Si l'identificateur de terminaison (terminationID) ne peut pas être déterminé de manière fiable, il envoie 422 (Erreur de syntaxe dans la commande) en tant que réponse d'action.

Si la fin d'une commande ne peut pas être déterminée de manière fiable, il renvoie 442 (erreur de syntaxe dans la commande) en tant que réponse à la dernière action qu'il peut analyser.

8.2.3 Transaction en cours (TransactionPending)

C'est le récepteur qui invoque la réponse de transaction en cours, qui indique que la transaction est en cours de traitement actif mais que ce traitement n'est pas terminé. Cette réponse sert à éviter que l'émetteur ne tire la conclusion que la demande de transaction a été perdue lorsque la transaction demande un certain temps pour s'exécuter.

TransactionPending(TransactionID { })

Le paramètre TransactionID doit être le même que celui de la demande TransactionRequest correspondante. Une propriété de racine (normalMGExecutionTime) est réglable par le contrôleur MGC afin d'indiquer l'intervalle pendant lequel la passerelle MG attend effectivement du contrôleur MGC une réponse à une transaction quelconque. Une autre propriété de racine (normalMGCExecutionTime) est réglable par le contrôleur MGC afin d'indiquer l'intervalle pendant lequel il convient que la passerelle MG attende du contrôleur MGC une réponse à une transaction quelconque. Les émetteurs peuvent recevoir plusieurs réponses de transaction en cours pour la même commande. Si une demande dupliquée est reçue pendant l'attente, le récepteur peut immédiatement envoyer une demande en attente ou continuer à attendre que son temporisateur déclenche une autre attente TransactionPending.

Une propriété de la terminaison racine (MGOriginatedPendingLimit) peut être fixée par le contrôleur MGC pour indiquer le nombre de transactions en cours qui peuvent être reçues en provenance de la passerelle MG. Lorsque la valeur indiquée par cette propriété est dépassée, la passerelle MG mettra fin au traitement de la transaction et renverra une réponse TransactionReply, afin que le contrôleur MGC ne pense pas que la transaction est fausse.

Une autre propriété de la terminaison racine (MGCOriginatedPendingLimit) peut être fixée par le contrôleur MGC pour indiquer le nombre de transactions en cours qui peuvent être reçues en provenance de lui. Lorsque la valeur indiquée par cette propriété est dépassée, le contrôleur MGC mettra fin au traitement de la transaction et renverra une réponse TransactionReply, afin que la passerelle MG ne pense pas que la transaction est fausse.

Le nombre limite de transactions en cours émises xxxOriginatedPendingLimit (MGOriginatedPendingLimit ou MGCOriginatedPendingLimit) peut être dépassé en raison soit du traitement d'une longue commande, soit d'une erreur (par exemple, une commande provoquant une boucle). Dans les deux cas, le récepteur de la demande de transaction initiale émettra une réponse TransactionReply avec un descripteur d'erreur comme paramètre de réponse, en rapport avec soit la longue commande incriminée, soit la commande ayant provoqué l'erreur. Les commandes suivantes de la transaction ne seront pas traitées. Erreur 506 (Nombre de transactions en cours dépassé).

NOTE – Afin d'éviter que le nombre limite de transactions en cours émises xxxOriginatedPendingLimit (MGOriginatedPendingLimit ou MGCOriginatedPendingLimit) soit dépassé en raison d'une erreur et que le récepteur de la demande de transaction initiale continue d'envoyer des transactions en cours, celui-ci devrait implémenter un mécanisme de protection de la gestion afin que soient déclenchées les mesures de

récupération appropriées. L'émetteur de la demande de transaction initiale peut suivre l'évolution du nombre de transaction en cours reçues et commencer à prendre des mesures correctives.

8.3 Messages

Plusieurs transactions peuvent être concaténées pour introduction dans un message. Les messages ont un en-tête contenant l'identité de l'émetteur. L'identificateur de message (MID, *message identifier*) est réglé sur un nom fourni (par exemple adresse de domaine/nom de domaine/nom de dispositif) de l'entité qui émet le message. Le nom de domaine est une valeur par défaut suggérée. Une entité H.248.1 (passerelle MG ou contrôleur MGC) doit employer de manière cohérente le même identificateur MID dans tous les messages qu'elle envoie pendant la durée de l'association en matière de commande avec son homologue (contrôleur MGC ou passerelle MG)

Chaque message contient un numéro de version désignant la version du protocole auquel le message est conforme. Les versions comportent un ou deux chiffres, en partant de la version 1. La version actuelle du protocole est la version 2.

Les transactions contenues dans un message sont traitées indépendamment. Aucun ordre n'est préétabli et il n'y a pas d'acquittement de message par l'application ou le protocole. Un message est avant tout un mécanisme de transport. Par exemple, il peut être répondu au message X contenant les demandes de transaction A, B et C par le message Y contenant les réponses à A et C et par le message Z contenant la réponse à B. De même, il peut être répondu au message L contenant une demande D et au message M contenant une demande E par le message N contenant les réponses aux messages D et E.

9 Transport

Le mécanisme de transport pour le protocole doit normalement permettre l'acheminement fiable de transactions entre un contrôleur MGC et une passerelle MG. Le transport doit rester indépendant des commandes particulières qui sont envoyées et il doit être applicable à tous les états applicatifs. Plusieurs transports sont définis pour le protocole dans les Annexes à la présente Recommandation et aux autres Recommandations H.248.x (par exemple, H.248.4 et H.248.5). D'autres modes de transport pourront être définis dans des Recommandations supplémentaires H.248.x. Concernant le transport du protocole au-dessus du protocole IP, les contrôleurs MGC doivent implémenter aussi bien le protocole TCP que le protocole UDP/ALF et une passerelle MG doit implémenter le protocole TCP ou le protocole UDP/ALF ou les deux.

La passerelle MG est pourvue d'un nom ou d'une adresse (tel qu'un nom de service DNS ou une adresse IP) d'un contrôleur MGC primaire et de zéro ou plus de zéro contrôleurs MGC secondaires (voir § 7.2.8). Cette adresse est celle que la passerelle MG utilisera pour envoyer des messages au contrôleur MGC. Si le protocole TCP ou UDP est utilisé comme le transport de protocole et si le port auquel la demande initiale de changement de service doit être envoyée n'est pas autrement connu, il convient d'envoyer la demande au numéro de port par défaut du protocole. Ce numéro de port est 2944 pour une opération codée au format alphanumérique ou 2945 pour une opération codée au format binaire, tant pour le protocole UDP que pour le protocole TCP. Le contrôleur MGC reçoit de la passerelle MG le message contenant la demande de changement de service et peut en déterminer l'adresse de la passerelle MG. Comme il est décrit dans le § 7.2.8, la passerelle MG ou le contrôleur MGC peut fournir une adresse dans le paramètre ServiceChangeAddress à laquelle des demandes de transaction subséquentes doivent être adressées, mais les réponses (y compris la réponse à la demande initiale de changement de service) doivent toujours être renvoyées à l'adresse source de la demande correspondante. Par exemple, dans les réseaux IP, cela correspond à l'adresse de la source dans l'en-tête IP et du numéro de port de la source dans l'en-tête TCP/UDP/SCTP.

9.1 Ordonnancement des commandes

La présente Recommandation ne prescrit pas que le protocole de transport sous-jacent garantisse la mise en séquence des transactions envoyées à une entité. Cette propriété tend à maximiser la ponctualité des actions mais elle comporte quelques inconvénients. Par exemple:

- les commandes Notify peuvent être retardées et arriver au contrôleur MGC après la transmission d'une nouvelle commande modifiant le descripteur d'événements;
- si une nouvelle commande est transmise avant qu'une précédente soit acquittée, rien ne garantit que cette commande antérieure sera exécutée avant la nouvelle.

Les contrôleurs MGC qui souhaitent garantir un fonctionnement cohérent de la passerelle MG peuvent appliquer les règles ci-après. Ces règles se rapportent aux commandes qui se trouvent dans des transactions différentes. Les commandes qui se trouvent dans la même transaction sont exécutées dans l'ordre (voir le paragraphe 8).

1) Lorsqu'une passerelle MG gère plusieurs terminaisons, les commandes relatives à ces différentes terminaisons peuvent être envoyées en parallèle, par exemple selon un modèle où chaque terminaison (ou groupe de terminaisons) est commandée par son propre processus ou par son propre fil d'exécution individuelle.

2) Sur une terminaison, il convient normalement qu'il y ait au plus une seule commande en instance (Add, Modify ou Move), à moins que les commandes en instance ne se trouvent dans la même transaction. Une commande Subtract peut cependant être émise à tout moment. En conséquence, une passerelle MG peut parfois recevoir une commande Modify qui s'applique à une terminaison déjà soustraite. De telles commandes doivent être ignorées et un code d'erreur doit être renvoyé.

3) Pour les transports qui ne garantissent la livraison en séquence des messages (à savoir UDP), il convient normalement qu'il y ait au niveau d'une terminaison donnée au plus une seule commande Notify en instance à un moment donné.

4) Dans certains cas, une commande Subtract implicitement ou explicitement remplacée par un caractère générique, qui s'applique à un groupe de terminaisons, peut se placer devant une commande Add en instance. Le contrôleur MGC devrait supprimer individuellement toutes les terminaisons pour lesquelles une commande Add était en instance au moment de la commande Subtract globale. Il convient également de ne pas envoyer de nouvelles commandes Add pour les terminaisons désignées par le remplacement par des caractères génériques (ou implicites dans un descripteur de multiplex) jusqu'à ce que la commande Subtract remplacée par un caractère générique soit acquittée.

5) Les commandes AuditValue et AuditCapability ne sont soumises à aucun séquencement.

6) La commande ServiceChange doit toujours être la première commande envoyée par une passerelle MG, comme défini par la procédure de redémarrage. Toute autre commande ou réponse doit être acheminée après cette commande ServiceChange.

Ces règles n'affectent pas le récepteur de la commande qui doit toujours répondre aux commandes.

9.2 Protection contre l'avalanche de redémarrage

Si un grand nombre de passerelles médias sont mises sous tension simultanément et si elles doivent toutes déclencher une transaction de changement de service, le contrôleur de passerelle média sera probablement submergé, ce qui provoque des pertes de message et un encombrement du réseau pendant la période critique du rétablissement du service. Afin d'éviter de telles avalanches, le comportement suivant est suggéré:

1) lorsqu'une passerelle MG est mise sous tension, elle doit armer un temporisateur de redémarrage de valeur aléatoire uniformément répartie entre 0 et un temps d'attente

maximal (MWD, *maximum waiting delay*). Il convient de veiller à éviter le synchronisme de production de nombres aléatoires entre plusieurs passerelles MG utilisant le même algorithme;

2) il convient que la passerelle MG attende soit l'expiration de cette temporisation soit la détection d'une activité d'utilisateur local, telle que par exemple une transition de décrochage sur une passerelle MG résidentielle;

3) lorsque la temporisation arrive à expiration ou lorsqu'une activité est détectée, il convient que la passerelle MG lance la procédure de redémarrage.

Celle-ci implique simplement que la passerelle MG garantisse que le premier message vu par le contrôleur MGC en provenance de cette passerelle sera un message de type ServiceChange informant le contrôleur MGC du redémarrage.

NOTE – La valeur du temps MWD est un paramètre de configuration qui dépend du type de passerelle MG. Le raisonnement suivant peut être utilisé afin de déterminer la valeur de ce temps dans les passerelles résidentielles.

Les contrôleurs MGC sont normalement dimensionnés de façon à absorber la charge de trafic à l'heure de pointe, pendant laquelle 10% des lignes en moyenne seront occupées à faire aboutir des communications dont la durée est normalement de 3 minutes. Le traitement d'une communication implique normalement 5 transactions à 6 transactions de contrôleur MGC entre chaque passerelle MG et le contrôleur MGC. Ce simple calcul montre que le contrôleur MGC est censé traiter 5 transactions à 6 transactions pour chaque terminaison, toutes les 30 minutes en moyenne. En d'autres termes, une transaction environ sera traitée toutes les 5 minutes à 6 minutes en moyenne dans chaque terminaison, ce qui permet d'estimer qu'une valeur raisonnable du temps MWD pour une passerelle résidentielle sera de 10 minutes à 12 minutes. En l'absence de configuration explicite, il y a lieu que les passerelles résidentielles adoptent une valeur de 600 secondes pour le temps MWD.

Le même raisonnement indique que la valeur du temps MWD devrait être beaucoup plus courte pour les passerelles de jonction ou pour les passerelles d'entreprise car elles gèrent un grand nombre de terminaisons dont le taux d'utilisation est aussi beaucoup plus élevé que 10% au cours de l'heure de pointe, une valeur typique étant 60%. Pendant l'heure de pointe, ces terminaisons sont censées contribuer à ce taux d'environ une transaction par minute à la charge du contrôleur MGC. Un algorithme rationnel consiste à rendre la valeur du temps MWD pour chaque terminaison "de jonction" six fois plus court que le temps MWD de chaque passerelle résidentielle et à rendre cette valeur inversement proportionnelle au nombre de terminaisons réinitialisées. Par exemple, le temps MWD devra être réglé à 2,5 secondes pour une passerelle qui gère une ligne au débit T1 ou à 60 ms pour une passerelle qui gère une ligne au débit T3.

10 Considérations relatives à la sécurité

Le présent paragraphe couvre la sécurité lors de l'utilisation du protocole dans un environnement IP.

10.1 Protection des connexions de protocole

Il est évident qu'un mécanisme de sécurité est nécessaire pour empêcher des entités non autorisées d'utiliser le protocole défini dans la présente Recommandation afin d'établir des communications non autorisées ou de perturber des communications autorisées. Le mécanisme de sécurité pour le protocole transporté par des réseaux en protocole IP est IPsec (RFC 2401 à RFC 2411).

L'en-tête AH (RFC 2402) offre l'authentification de l'origine des données, l'intégrité du mode sans connexion et la protection facultative contre le redéfilement de messages transmis entre la passerelle MG et le contrôleur MGC. L'en-tête ESP (RFC 2406) assure au besoin la confidentialité des messages. Par exemple, le service de chiffrement de charge utile par encapsulation IP (ESP)

doit être demandé si les descriptions de session sont utilisées pour acheminer des clés de session, comme défini dans le protocole SDP.

Les implémentations du protocole défini dans la présente Recommandation utilisant l'en-tête du protocole ESP DOIVENT être conformes à la section 5 de RFC 2406 qui définit un ensemble minimal d'algorithmes pour le contrôle d'intégrité et le cryptage. De même, les implémentations utilisant l'en-tête AH DOIVENT être conformes à la section 5 de RFC 2402 qui définit un ensemble minimal d'algorithmes pour le contrôle d'intégrité au moyen de clés manuelles.

Les implémentations DEVRAIENT utiliser le chiffrement IKE RFC 2409 pour permettre des options de calcul de clé plus robustes. Les implémentations utilisant le chiffrement IKE DEVRAIENT prendre en charge l'authentification par signatures à codage RSA et le chiffrement de clés publiques par codage RSA.

10.2 Système provisoire d'en-tête AH

L'implémentation du mécanisme IPsec nécessite que l'en-tête AH ou ESP soit inséré immédiatement après l'en-tête IP, ce qui ne peut pas être fait aisément au niveau de l'application. C'est donc un problème de déploiement pour le protocole défini dans la présente Recommandation lorsque l'implémentation du réseau sous-jacent ne prend pas en charge le mécanisme IPsec.

A titre de solution intérimaire, un en-tête AH facultatif est défini à l'intérieur de l'en-tête de protocole de H.248.1. Les champs de cet en-tête sont exactement les mêmes que les champs SPI, SEQUENCE NUMBER et DATA qui sont définis dans RFC 2402. La sémantique des champs d'en-tête est la même que celle du "mode de transport" de RFC 2402, sauf pour le calcul de la valeur de contrôle d'intégrité (ICV, *integrity check value*). Dans le mécanisme IPsec, la valeur ICV est calculée sur le paquet IP entier, y compris son en-tête IP. Cela empêche la simulation des adresses IP. Pour conserver la même capacité, le calcul de la valeur ICV doit normalement être effectué sur toutes les transactions (concaténées) dans le message préfixé d'un en-tête IP synthétisé à partir d'une adresse IP de source sur 32 bits, d'une adresse de destination sur 32 bits et d'un port de destination UDP de 16 bits codé sous forme de 20 chiffres hexadécimaux. Lorsque le mécanisme intérimaire d'en-tête AH est employé alors que le protocole TCP constitue la couche Transport, l'accès UDP ci-dessus devient l'accès TCP et toutes les autres opérations sont les mêmes.

Les implémentations du protocole H.248.1 DOIVENT implémenter le mécanisme IPsec lorsque le système d'exploitation et le réseau de transport sous-jacents prennent en charge ce mécanisme. Les implémentations du protocole utilisant la version IPv4 DOIVENT implémenter le schéma AH intérimaire. Ce système provisoire ne DOIT PAS cependant être utilisé lorsque la couche Réseau sous-jacente prend en charge le mécanisme IPsec. Les implémentations en version IPv6 sont censées prendre en charge le mécanisme IPsec et ne DOIVENT PAS utiliser le système provisoire d'en-tête AH.

Toutes les implémentations du mécanisme intérimaire d'en-tête AH DOIVENT être conformes à la section 5 de RFC 2402 qui définit un ensemble minimal d'algorithmes pour le contrôle de l'intégrité au moyen de clés manuelles.

Le système provisoire d'en-tête AH n'assure pas la protection contre les interceptions illicites, ce qui empêche des tierces parties de surveiller les connexions établies par une terminaison donnée. Il n'assure pas non plus de protection contre les attaques par redéfilement. Ces procédures ne protègent pas nécessairement contre les attaques par déni de service de passerelles MG ou de contrôleurs MGC à comportement erroné. Elles assureront cependant une identification de ces entités à comportement erroné, qui devront alors être privées de leur autorisation au moyen de procédures de maintenance.

10.3 Protection des connexions de médias

Le protocole permet au contrôleur MGC de fournir aux passerelles MG des "clés de session" pouvant servir à chiffrer les messages audio et à protéger contre les interceptions illicites.

Un problème propre aux réseaux en mode paquet est "l'intervention non contrôlée". Cette attaque peut être exécutée par un aiguillage des paquets médias vers l'adresse IP et l'accès UDP utilisés par une connexion. Si aucune protection n'est implémentée, les paquets doivent être décomprimés et les signaux doivent être exécutés du côté "ligne".

Une protection de base contre cette attaque consiste à n'accepter que les paquets provenant de sources connues, en vérifiant par exemple que l'adresse source IP et l'accès source UDP correspondent aux valeurs annoncées dans le descripteur Remote. Ce procédé présente deux inconvénients: il ralentit l'établissement de la connexion et il peut être contourné par une simulation de source:

- pour activer la protection fondée sur l'adresse, le contrôleur MGC doit toujours obtenir de la passerelle MG de sortie la description de la session distante puis la transmettre à la passerelle MG d'entrée. Cela nécessite au moins un aller et retour dans le réseau et pose un dilemme: soit laisser l'appel s'établir sans attendre la fin de l'aller et retour avec le risque, par exemple, d'une "mutilation" d'une annonce distante, soit attendre la fin de l'aller et retour et accepter des procédures d'établissement d'appel plus lentes;
- la simulation de source n'est effective que si l'attaquant peut obtenir des paires valides d'adresses et d'accès de source et de destination, par exemple par écoute d'une fraction du trafic. Pour combattre la simulation de source, l'on peut essayer de contrôler tous les points d'accès au réseau; mais cela est très difficile à réaliser en pratique.

Une autre solution pour vérifier l'adresse de source consiste à chiffrer et à authentifier les paquets au moyen d'une clé secrète qui est acheminée pendant la procédure d'établissement d'appel. Cela ne ralentira pas celui-ci et apportera une bonne protection contre la simulation de source.

11 Interface de commande MG-MGC

L'association de commande entre passerelle MG et contrôleur MGC est créée lors d'un démarrage à froid de la passerelle MG puis annoncée par un message de changement de service. Mais cette association peut être modifiée par des événements ultérieurs, comme des défaillances ou des événements de service manuel.

NOTE – Bien qu'il ne dispose pas d'un mécanisme explicite pour prendre en charge de multiples contrôleurs MGC commandant une passerelle MG physique, le protocole a été conçu de façon à prendre en charge les multiples passerelles MG logiques (contenues dans une même passerelle MG physique) qui peuvent être associées à différents contrôleurs MGC.

11.1 Multiples passerelles MG virtuelles

Une passerelle MG physique peut être partitionnée en une ou plusieurs passerelles MG virtuelles. Une passerelle MG virtuelle se compose d'un paquetage de terminaisons physiques statiquement partitionnées ou d'ensembles de terminaisons éphémères. Une terminaison physique est commandée par un seul contrôleur MGC. Le modèle n'exige pas que d'autres ressources que les terminaisons soient attribuées statiquement. Le mécanisme d'attribution des terminaisons aux passerelles MG virtuelles est une méthode de gestion hors du domaine d'application de la présente Recommandation. Chacune des passerelles MG virtuelles apparaît au contrôleur MGC comme un client de passerelle MG complet.

Une passerelle MG physique ne peut avoir qu'une seule interface avec le réseau, qui doit toujours être partagée entre passerelles MG virtuelles. Dans un tel cas, la terminaison du côté paquets/cellules est partagée. Il convient toutefois de noter qu'en exploitation de telles interfaces

nécessitent une instance éphémère de la terminaison à créer dans chaque flux, ce qui simplifie le partage de la terminaison. Ce mécanisme implique toutefois une complication, à savoir que la passerelle MG doit toujours savoir lequel de ses contrôleurs MGC directeurs doit être averti de l'apparition d'un événement à l'interface.

En exploitation normale, la passerelle MG virtuelle sera appelée par le contrôleur MGC à créer des flux dans le réseau (si elle est du côté départ) ou à attendre des flux dans le réseau (si elle est du côté arrivée), de sorte qu'aucune confusion n'apparaîtra. Si cependant un événement imprévu se produit, la passerelle MG virtuelle doit savoir ce qu'il faut faire en ce qui concerne les ressources physiques qu'elle commande.

Si la reprise sur l'événement nécessite la manipulation d'un état de l'interface physique, il convient qu'un seul contrôleur MGC l'effectue. La solution de ces problèmes consiste à autoriser tout contrôleur MGC à créer des descripteurs d'événement qui recevront notification de ces événements. Mais un seul contrôleur MGC pourra avoir l'accès en lecture/écriture aux propriétés de l'interface physique, tous les autres contrôleurs MGC n'ayant qu'un accès en lecture. Le mécanisme de gestion est utilisé pour désigner le contrôleur MGC qui possède la capacité de lecture/écriture et qui est appelé contrôleur MGC maître.

Chaque passerelle MG virtuelle possède sa propre terminaison racine. Le plus souvent, les valeurs des propriétés de cette terminaison racine sont réglables indépendamment par chaque contrôleur MGC. Lorsqu'il ne peut y avoir qu'une seule valeur, ce paramètre est en lecture seulement pour tous les contrôleurs MGC sauf le contrôleur maître.

Une commande ServiceChange ne peut être appliquée qu'à une terminaison ou qu'à un ensemble de terminaisons partitionné(e) selon la passerelle MG virtuelle ou créé(e) (dans le cas de terminaisons éphémères) par cette passerelle MG virtuelle.

11.2 Démarrage à froid

Une passerelle MG est préconfigurée, par un mécanisme de gestion extérieur au domaine d'application de la présente Recommandation, avec un contrôleur MGC primaire et (facultativement) une liste ordonnée de contrôleurs MGC secondaires. Lors du démarrage à froid de la passerelle MG, celle-ci enverra à son contrôleur MGC primaire une commande ServiceChange avec méthode de type "redémarrage" (valeur "restart") au sujet de la terminaison racine. Si le contrôleur MGC accepte la passerelle MG, il envoie une réponse de transaction ne contenant pas de paramètre ServiceChangeMgcId. S'il n'accepte pas l'enregistrement de la passerelle MG, il envoie une réponse de transaction, fournissant l'adresse d'un autre contrôleur MGC à contacter, en incluant un paramètre ServiceChangeMcgId.

Si la passerelle MG reçoit une réponse de transaction qui comporte un paramètre ServiceChangeMcgId, elle envoie une commande ServiceChange au contrôleur MGC indiqué dans le paramètre ServiceChangeMcgId. Elle continue ce processus jusqu'à ce qu'elle obtienne un contrôleur MGC qui accepte son enregistrement; sinon, sa recherche de réponse, en provenance du contrôleur MGC primaire ou d'un successeur désigné, échoue. Dans ce dernier cas, la passerelle MG essaie, dans l'ordre, ses contrôleurs MGC secondaires préconfigurés. Si la passerelle MG ne peut pas établir une relation de commande avec un contrôleur MGC quelconque, elle doit attendre pendant un temps aléatoire comme décrit au § 9.2 et ensuite commencer à contacter de nouveau son contrôleur MGC primaire et, le cas échéant, ses contrôleurs MGC secondaires.

Il est possible qu'une réponse à un changement de service avec Restart soit perdu et que la passerelle MG reçoive une commande avant la réception de la réponse au changement de service. La passerelle MG doit émettre une erreur 505 (Demande de transaction reçue avant d'avoir reçu la réponse au changement de service).

11.3 Négociation de version de protocole

Une commande ServiceChange provenant d'une passerelle MG qui est enregistrée auprès du contrôleur MGC doit contenir le numéro de version du protocole pris en charge par la passerelle dans le paramètre ServiceChangeVersion. Quelle que soit la version indiquée dans le paramètre ServiceChangeVersion, le message contenant la commande sera codé comme un message de la version 1. A la réception d'un tel message, si le contrôleur MGC ne prend en charge qu'une version inférieure, le contrôleur MGC doit envoyer une réponse ServiceChangeReply avec la version inférieure et, par la suite, tous les messages entre la passerelle MG et le contrôleur MGC doivent être conformes à la version inférieure du protocole. Si la passerelle MG ne peut pas se conformer et si elle a établi une connexion de transport au contrôleur, il convient que la passerelle ferme cette connexion. Dans tous les cas, il convient qu'elle rejette toutes les demandes subséquentes provenant du contrôleur MGC avec l'erreur 406 (Version non prise en charge).

Si le contrôleur MGC prend en charge une version supérieure à celle de la passerelle MG, il est capable de prendre en charge la version inférieure proposée par la passerelle MG, le contrôleur MGC doit envoyer une réponse ServiceChangeReply avec la version inférieure et, par la suite, tous les messages entre la passerelle MG et le contrôleur MGC doivent être conformes à la version inférieure du protocole. Si le contrôleur MGC ne peut pas se conformer, il doit rejeter l'association, avec l'erreur 406 (Version non prise en charge).

La négociation de la version de protocole peut également se produire aux changements de service "handoff" (transfert) et "failover" (reprise sur défaillance).

Lors de l'extension du protocole avec de nouvelles versions, il convient de suivre les règles ci-après:

1) il convient de ne pas modifier les éléments de protocole existants, à savoir procédures, paramètres, descripteur, propriété, valeur, à moins qu'il ne soit nécessaire de corriger une erreur de protocole ou de changer le fonctionnement du service qui est en cours de prise en charge par le protocole;

2) il convient de ne pas changer la sémantique d'une commande, d'un paramètre, d'un descripteur, d'une propriété ou d'une valeur;

3) il convient de ne pas modifier les règles établies de formatage et de codage des messages et des paramètres;

4) lorsque des éléments d'information sont obsolètes, on peut les marquer comme n'étant pas utilisés. Toutefois, l'identificateur pour cet élément d'information est marqué comme étant réservé. Ainsi, on ne peut pas l'utiliser dans de futures versions.

11.4 Défaillance d'une passerelle MG

Si une passerelle MG subit une défaillance mais reste capable d'envoyer un message au contrôleur MGC, elle envoie une commande ServiceChange avec une méthode appropriée (de valeur "graceful" ou "forced") puis elle spécifie l'identificateur de terminaison racine. Lorsqu'elle est remise en service, elle envoie une commande ServiceChange avec une méthode de type "Restart".

L'autorisation donnée au contrôleur MGC d'envoyer des messages en double à deux passerelles MG permet de gérer des paires de passerelles MG ayant la capacité de reprise sur défaillance par redondance réciproque. Seule la passerelle MG en service doit accepter ou rejeter des transactions. En cas de reprise sur défaillance, la passerelle MG primaire envoie une commande ServiceChange avec une méthode de type "reprise sur défaillance" et une cause de type "défaillance imminente de passerelle MG". Le contrôleur MGC utilise alors la passerelle MG secondaire comme passerelle active. Lorsque l'état d'erreur est réparé, la passerelle MG en service peut envoyer une commande ServiceChange avec une méthode de type "Restart".

NOTE – Des passerelles à reprises sur défaillance redondantes nécessitent un transport fiable, parce que le protocole ne donne pas les moyens à une passerelle secondaire utilisant le verrouillage au niveau de la couche Application ALF d'accuser réception des messages envoyés par le contrôleur MGC.

11.5 Défaillance d'un contrôleur MGC

Si la passerelle MG détecte une défaillance de son contrôleur MGC, elle tente d'établir un contact avec le prochain contrôleur MGC figurant sur sa liste préconfigurée. Elle commence ses essais par le début de la liste (c'est-à-dire par le contrôleur MGC primaire, à moins que celui-ci ne soit en panne, auquel cas elle commence par son premier contrôleur MGC secondaire). Elle envoie un message ServiceChange avec une méthode de type "reprise sur défaillance" et une raison de type "défaillance imminente du contrôleur MGC". Si elle n'est pas en mesure d'établir une relation de commande avec un quelconque contrôleur MGC, elle attendra pendant un temps de durée aléatoire, tel que décrit au § 9.2 et reprendra ses contacts avec son contrôleur primaire et (si besoin est) ses contrôleurs secondaires. Lorsqu'elle contacte les contrôleurs ayant effectués les contrôles précédents, elle envoie un message ServiceChange avec la méthode "Disconnected".

En cas de défaillance partielle ou pour des raisons de maintenance manuelle, un contrôleur MGC peut souhaiter ordonner à ses passerelles MG contrôlées d'utiliser un autre contrôleur MGC. A cette fin, il envoie à la passerelle MG une commande ServiceChange avec une méthode de type "Handoff" ainsi que, dans un paramètre ServiceChangeMgcId, la désignation du contrôleur qui le remplacera. Si une méthode de type "Handoff" est prise en charge, la passerelle enverra au contrôleur MGC ainsi désigné un message ServiceChange avec une méthode de type "Handoff" et une raison de type "changement ordonné par le contrôleur MGC". Si la passerelle ne parvient pas à obtenir une réponse du contrôleur MGC désigné, elle doit se comporter comme si son contrôleur MGC avait subi une défaillance et elle doit commencer à prendre contact avec ses contrôleurs MGC secondaires comme indiqué dans l'alinéa précédent. Si la passerelle ne peut pas établir une relation de commande avec un quelconque contrôleur MGC, elle doit attendre pendant un temps aléatoire comme il est décrit dans le § 9.2 et ensuite commencer à contacter de nouveau ses contrôleurs primaires et, le cas échéant, secondaires.

Aucune recommandation n'est donnée sur la façon dont les contrôleurs MGC impliqués dans le transfert conservent les informations d'état; ce sujet est considéré ne pas s'inscrire dans le cadre de la présente Recommandation. Le contrôleur MGC et la passerelle MG peuvent suivre les étapes ci-après lorsque le transfert se produit. Lorsque le contrôleur MGC entreprend un transfert, celui-ci doit normalement être transparent aux opérations effectuées dans la passerelle MG. Les transactions peuvent être exécutées dans un ordre quelconque et peuvent être en cours lorsque la commande ServiceChange est exécutée. En conséquence, les commandes en cours se poursuivent et des réponses à toutes les commandes provenant du contrôleur MGC initial doivent être envoyées aux adresse de transport d'où celles-ci avaient été envoyées. Si la relation de service avec le contrôleur MGC expéditeur s'est achevée, les réponses devraient être ignorées. La passerelle MG peut recevoir des réponses concernant des transactions en cours envoyées par le nouveau contrôleur MGC. Aucun nouveau message ne doit être envoyé du nouveau contrôleur MGC jusqu'à ce que l'association de commande soit établie. Les demandes de transaction répétées doivent être dirigées vers le nouveau contrôleur MGC. La passerelle doit conserver l'état de toutes les terminaisons et de tous les contextes.

Il est possible que le contrôleur MGC puisse être implémenté de telle façon qu'il soit, en cas de défaillance, remplacé par un nouveau contrôleur MGC en service dont l'identité est la même que celle du contrôleur défaillant. Dans un tel cas, le paramètre ServiceChangeMgcId est spécifié par la précédente valeur et la passerelle MG doit se comporter comme si la valeur avait été modifiée et envoyer un message de changement de service (ServiceChange), comme ci-dessus.

Des paires de contrôleurs MGC possédant la capacité de reprise sur défaillance par redondance peuvent, au moyen du mécanisme ci-dessus, signaler les reprises sur défaillance aux passerelles MG contrôlées.

12 Définition des paquetages

Le mécanisme primaire d'extension est celui des paquetages. Ceux-ci définissent des propriétés, des événements, des signaux et des statistiques supplémentaires qui peuvent se produire dans des terminaisons.

Les paquetages définis par le groupe IETF apparaîtront dans des commentaires RFC distincts.

Les paquetages définis par l'UIT-T peuvent apparaître dans les Recommandations applicables (par exemple en tant que Recommandations de la série H.248.x).

1) Il y a lieu de spécifier un document public ou un document de forum de normalisation pouvant être référencé comme étant le document décrivant chaque paquetage conforme aux directives ci-dessus.

2) Ce document doit spécifier la version du paquetage qu'il décrit.

3) Ce document devra être disponible sur un serveur Internet public et devra avoir un URL stable. Ce site devra offrir la possibilité de formuler des observations et les réponses appropriées devront être renvoyées.

12.1 Directives pour la définition des paquetages

Les paquetages définissent des propriétés, des événements, des signaux et des statistiques.

Les paquetages peuvent également définir des codes d'erreur conformément aux directives fournies dans le § 14.2. Il s'agit d'une question de commodité documentaire: la documentation des paquetages est soumise à l'autorité IANA en soutien à l'enregistrement des codes d'erreur. Si un paquetage est modifié, il n'est pas nécessaire de fournir à l'autorité IANA une nouvelle référence de document à l'appui des codes d'erreur, à moins que la description du code d'erreur ne soit elle-même modifiée.

Les noms de toutes les structures ainsi définies doivent se composer de l'identificateur de paquetage (qui désigne celui-ci de manière unique) et de l'identificateur de l'item (désignant de manière unique l'item contenu dans le paquetage). Dans un codage alphanumérique, ces deux identificateurs doivent être séparés par un caractère barre oblique ("/"). Par exemple, l'expression "togen/playtone" désigne le codage alphanumérique pour se référer au signal de tonalité d'exécution contenu dans le paquetage de production de tonalités.

Les paragraphes suivants décrivent la composition d'un paquetage:

12.1.1 Paquetage

Description globale du paquetage, spécifiant:

nom du paquetage: élément descriptif seulement

identificateur du paquetage: identificateur

description:

version:

une nouvelle version d'un paquetage ne peut qu'ajouter des propriétés, événements, signaux et statistiques supplémentaires ainsi que d'éventuelles valeurs nouvelles pour un paramètre existant qui est décrit dans le paquetage initial. Aucune suppression ou modification n'est autorisée. Une version est un nombre entier dans la plage de 1 à 99.

Conçu uniquement pour être étendu (en option): oui

cela indique que le paquetage a été expressément conçu pour être étendu par d'autres paquetages, ne devant pas être référencés directement. Par exemple, le paquetage peut n'assurer aucune fonction lui-même ou être en soi dénué de sens. La passerelle MG NE DOIT PAS publier cet identificateur de paquetage lorsqu'elle établit le rapport des paquetages.

Extension (facultative): descripteur de paquetage existant

un paquetage peut constituer une extension d'un paquetage existant. La version du paquetage initial doit être spécifiée. Lorsqu'un paquetage développe un autre paquetage, il ne peut qu'ajouter des propriétés, événements, signaux et statistiques supplémentaires ainsi que d'éventuelles valeurs nouvelles pour un paramètre existant qui est décrit dans le paquetage initial. Un paquetage étendu ne doit pas redéfinir ou recouvrir un identificateur défini dans le paquetage initial et dans les paquetages qu'il a pu étendre (niveaux multiples d'extension). Si donc la version 1 du paquetage B étend la version 1 du paquetage A, la version 2 de B ne pourra pas étendre la version 2 de A si celle-ci définit un nom qui existe déjà dans la version 1 de B.

12.1.2 Propriétés

Propriétés définies par le paquetage, spécifiant:

nom de la propriété: élément descriptif seulement

identificateur de propriété: identificateur

description:

type de paramètre: à choisir parmi les suivants:

booléen

chaîne: chaîne en format UTF-8

chaîne d'octets: nombre d'octets. Se reporter aux Annexes A et B.3 en ce qui concerne le codage

entier: entier signé de 4 octets

double: entier signé de 8 octets

caractère: codage de chaque lettre en unicode UTF-8, éventuellement sur plusieurs octets

énumération: liste de valeurs uniques à choisir (se reporter au § 12.3)

sous-liste: liste de plusieurs valeurs tirées d'une liste. Ce type de sous-liste DOIT aussi être spécifié. Le type sera choisi parmi les types indiqués dans le présent paragraphe (à l'exception du type sous-liste). Par exemple, la sous-liste peut être de type sous-liste d'énumération. Le codage des sous-listes est spécifié aux Annexes A et B.3.

Valeurs possibles:

un paquetage DOIT indiquer un ensemble particulier de valeurs ou donner une description de la manière de déterminer des valeurs. Il DOIT aussi spécifier une valeur par défaut ou le comportement par défaut lorsque la valeur n'est pas présente dans son descripteur. Par exemple, un paquetage peut indiquer que les procédures liées à la propriété sont suspendues lorsque la valeur est omise. Une valeur par défaut (mais non les procédures) peut être spécifiée comme étant susceptible d'être fournie.

Définies dans:

le descripteur H.248.1 définissant la propriété. Le descripteur LocalControl définit les propriétés qui dépendent du flux. Le descripteur TerminationState définit les propriétés qui sont indépendantes du flux. Ces descripteurs devraient couvrir les cas les plus courants mais il est possible que des propriétés soient définies dans d'autres descripteurs.

Caractéristiques: en lecture/en écriture ou les deux et (facultativement) globale:

indique si une propriété est en lecture seule ou en lecture et écriture et si elle est globale. Si l'attribut "Global(e)" est omis, la propriété n'est pas globale. Si une propriété est déclarée globale, sa valeur est partagée par toutes les terminaisons réalisant le paquetage.

12.1.3 Evénements

Evénements définis par le paquetage, spécifiant:

nom de l'événement: élément descriptif seulement

identificateur d'événement: identificateur

description:

paramètres du descripteur d'événements:

paramètres utilisés par le contrôleur MGC pour configurer l'événement, contenus dans le descripteur d'événements. Se reporter au § 12.2.

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor):

paramètres qui sont renvoyés au contrôleur MGC dans des demandes Notify et dans des réponses à des demandes de commande envoyées par les contrôleurs MGC auditant le descripteur ObservedEvents, et qui se trouvent dans le descripteur ObservedEvents. Se reporter au § 12.2.

12.1.4 Signaux

Signaux définis par le paquetage, spécifiant:

nom de signal: élément descriptif seulement

identificateur de signal: identificateur utilisé dans un descripteur de signaux

description:

type de signal: à choisir parmi les suivants:

OO (commuté)

TO (temporisé)

BR (bref)

NOTE – Le type de signal peut être défini de façon qu'il dépende de la valeur d'un ou de plusieurs paramètres. Le paquetage DOIT indiqué un type de signal par défaut. Si ce type par défaut est TO, le paquetage DOIT spécifier une durée par défaut qui peut être assurée. Une durée par défaut est dénuée de sens pour le type BR.

Durée: exprimée en centièmes de secondes

Paramètres additionnels: voir § 12.2.

12.1.5 Statistiques

Statistiques définies par le paquetage, spécifiant:

nom de statistique: élément descriptif seulement

identificateur de statistique: identificateur

identificateur utilisé dans un descripteur de statistique

description

unités: unité de mesure, par exemple millisecondes, paquets

12.1.6 Procédures

Directives additionnelles concernant l'emploi du paquetage.

12.2 Directives de définition des paramètres relatifs aux événements et signaux

Nom de paramètre: élément descriptif seulement

identificateur de paramètre: identificateur. L'identificateur alphanumérique des paramètres Events et Signals ne commenceront pas par "EPA" et "SPA" respectivement. L'identificateur alphanumérique de paramètre ne sera pas non plus "ST", "Stream", "SY", "SignalType", "DR", "Duration", "NC", "NotifyCompletion", "KA", "Keepactive", "EB", "Embed", "DM" ou "DigitMap".

Type de paramètre: à choisir parmi les suivants:

booléen

chaîne: UTF-8 octet string

chaîne d'octets: nombre d'octets. Se reporter aux Annexes A et B.3 en ce qui concerne le codage.

entier: entier signé de 4 octets

double: entier signé de 8 octets

caractère: codage de chaque lettre en unicode UTF-8, éventuellement sur plusieurs octets

énumération: liste de valeurs uniques à choisir (voir § 12.3)

sous-liste: liste de plusieurs valeurs tirées d'une liste (non prise en compte pour les statistiques). Ce type de sous-liste DOIT aussi être spécifié. Le type sera choisi parmi les types indiqués dans le présent paragraphe (à l'exception du type sous-liste). Par exemple, la sous-liste peut être de type sous-liste d'énumération. Le codage des sous-listes est spécifié aux Annexes A et B.3.

Valeurs possibles:

un paquetage DOIT indiquer un ensemble particulier de valeurs ou donner une description de la manière de déterminer des valeurs. Il DOIT aussi spécifier une valeur par défaut ou le comportement par défaut lorsque la valeur n'est pas présente dans son descripteur. Par exemple, un paquetage peut indiquer que les procédures liées au paramètre sont suspendues lorsque la valeur est omise. Une valeur par défaut (mais non les procédures) peut être spécifiée comme étant susceptible d'être fournie.

Description:

12.3 Listes

Les valeurs possibles des paramètres incluent les énumérations. Des énumérations peuvent être définies dans une liste. Il est recommandé que la liste soit enregistrée par l'autorité IANA de façon

que les paquetages qui étendent cette liste puissent être définis sans qu'il y ait lieu de se préoccuper de conflits de noms.

12.4 Identificateurs

Les identificateurs codés en mode alphanumérique doivent être des chaînes de 64 caractères au plus, ne contenant pas d'espaces, commençant par un caractère alphabétique et contenant des caractères alphanumériques et/ou des chiffres avec possibilité d'insertion du caractère spécial de soulignement ("_").

Les identificateurs codés en mode binaire ont une longueur de 2 octets.

Les valeurs alphanumériques et les valeurs binaires doivent toutes deux être spécifiées pour chaque identificateur, y compris les identificateurs utilisés comme valeurs dans les types énumérés.

12.5 Enregistrement de paquetage

Un paquetage peut être enregistré auprès de l'autorité IANA pour des raisons d'interopérabilité. Se reporter au paragraphe 13 pour des considérations relatives à l'autorité IANA.

13 Définition des profils

Des profils peuvent être spécifiés afin de mieux définir l'emploi du protocole H.248.1 et les fonctionnalités qui sont prises en charge par une passerelle MG. Le profil lui-même spécifie les options associées avec la Rec. UIT-T H.248.1 qui ont été employées. Par exemple: le transport et les paquetages utilisés pour une application.

Un profil est identifié par un nom (enregistré auprès de l'autorité IANA) et une version. Le nom consistera en une chaîne d'au plus 64 caractères. La version peut varier de 1 à 99.

Le profil lui-même est un document qui indique les options pour une application particulière. Aucun format n'est fixé pour ce document. Seule est obligatoire la présence d'une section indiquant le nom et la version et d'un résumé du profil.

Tandis que les deux premiers éléments sont obligatoires, il conviendrait d'envisager d'inclure les éléments suivants:

- identification du profil: nom et version du profil envoyé dans la commande de modification du service;
- résumé: description du profil;
- conventions relatives à la dénomination:
  - conventions relatives à la dénomination MGC/MG: adressage associé aux noms du contrôleur MGC/de la passerelle MG;
  - noms des terminaisons: structure de l'identité des terminaisons;
  - noms des scripts de numérotation;
- descripteur de topologie: ce profil emploie-t-il le descripteur de topologie?;
- timbres horodateurs: éléments indiquant l'emploi de timbres horodateurs dans les commandes ServiceChange et/ou Notify;
- temporisateurs de transaction: éléments indiquant les valeurs des temporisateurs de transaction;
- transport: élément indiquant les transports conformes à la série de Recommandations H.248, qui sont pris en charge par le profil;
- codage: élément indiquant le codage pris en charge par le profil;

- prise en charge obligatoire du protocole SDP et des éléments d'information de l'Annexe C: éléments indiquant les attributs du protocole SDP et les éléments d'information de l'Annexe C, devant être pris en charge;
- paquetages: éléments indiquant les paquetages pris en charge dans ce profil;

  obligatoire: attribut indiquant les paquetages qui doivent être pris en charge dans ce profil;

  en option: attribut indiquant les paquetages qui peuvent être pris en charge dans ce profil;

  informations relatives à la fourniture des paquetages: éléments indiquant les valeurs des propriétés dont il est spécifié qu'elles sont assurées (par exemple, noms et nombre de cycles pour une annonce conforme à la Rec. UIT-T H.248.7);
- sécurité: élément indiquant les mécanismes de sécurité employés;
- procédures: éléments indiquant les procédures associées à ce profil.

14 Considérations relatives à l'autorité IANA

14.1 Paquetages

Les prescriptions suivantes DOIVENT être observées pour enregistrer un paquetage auprès de l'autorité IANA:

1) pour chaque paquetage, une chaîne nominative unique, un numéro de série unique et un numéro de version doivent être enregistrés. La chaîne nominative est utilisée dans le cadre du codage en mode alphanumérique. Le numéro de série est utilisé dans le cadre du codage en mode binaire. Les numéros de série 0x8000 à 0xFFFF sont réservés à l'usage privé. Le numéro de série 0 est réservé;

2) un nom de contact, une adresse de courrier électronique et une adresse postale doivent être spécifiés pour ce contact. Ces informations de contact doivent être mises à jour, le cas échéant, par l'organisation chargée de la définition;

3) une référence à un document décrivant le paquetage, qui doit être public, doit être indiquée:

   Ce document doit spécifier la version du paquetage qu'il décrit.

   Si le document est public, il devra être disponible sur un serveur Internet public et avoir un URL stable. Ce site devra offrir la possibilité de formuler des observations et les réponses appropriées devront être renvoyées;

4) les paquetages enregistrés par des organisations de normalisation autres que celles qui sont reconnues doivent avoir une longueur de chaîne nominative au moins égale à 8 caractères;

5) tous les autres noms de paquetage sont servis dans l'ordre d'arrivée si toutes les autres conditions sont satisfaites.

14.2 Codes d'erreur

Les prescriptions suivantes DOIVENT être observées pour enregistrer un code d'erreur auprès de l'autorité IANA:

1) un numéro d'erreur et une seule ligne (chaîne de 80 caractères au plus) sont enregistrés pour chaque erreur;

2) une description complète des conditions dans lesquelles l'erreur est détectée doit être incluse dans un document disponible au niveau public. Cette description doit être suffisamment claire pour différencier l'erreur de tout autre code d'erreur existant;

3) ce document devra être disponible sur un serveur Internet public et devra avoir un URL stable;

4) les numéros d'erreur enregistrés par des organismes de normalisation reconnus doivent se composer de 3 ou 4 caractères;

5) les numéros d'erreur enregistrés par tous les autres organismes ou individus doivent se composer de 4 caractères;

6) un numéro d'erreur ne doit pas être redéfini ni modifié sauf par l'organisation ou l'individu qui l'a défini initialement, ou par ses successeurs ou commissionnaires.

14.3 Raisons de changement de service

Les prescriptions suivantes DOIVENT être observées pour enregistrer une raison de changement de service auprès de l'autorité IANA:

1) un code unique de raison, constitué d'une seule phrase de 80 caractères au plus, est enregistré pour chaque raison;

2) une description complète des conditions dans lesquelles la raison est utilisée doit être incluse dans un document disponible au niveau public. Cette description doit être suffisamment claire pour différencier la raison de tout autre code de raison existant;

3) ce document devra être disponible sur un serveur Internet public et devra avoir un URL stable.

14.4 Profils

Les considérations suivantes DOIVENT être satisfaites aux fins de l'enregistrement auprès de l'autorité IANA:

1) un seul nom de chaîne et un seul numéro de version (la version peut être omise lorsque le nom de profil contient un caractère générique) sont enregistrés pour chaque profil;

2) le nom, l'adresse électronique et l'adresse postale d'une personne pouvant être contactée seront indiqués. Ces informations seront mises à jour, si besoin est, par l'organisme les ayant données;

3) les noms des profils enregistrés par des organismes autres que les organismes de normalisation agréés doivent comporter six caractères au moins;

4) les noms de profils se terminant par un caractère générique "*" doivent être acceptés si les six premiers caractères sont entièrement spécifiés. On suppose que l'organisme ayant reçu le nom de profil gérera l'espace de ce nom, associé au caractère générique. L'autorité IANA ne délivrera pas d'autre nom de profil dans le domaine "nom*".

Tous les autres noms de profil seront attribués selon l'ordre d'arrivée lorsque l'ensemble des autres conditions sont réunies.

Annexe A

Codage binaire du protocole

La présente annexe spécifie la syntaxe des messages au moyen de la notation définie dans la Rec. UIT-T X.680: *Technologies de l'information – Notation de syntaxe abstraite numéro un*: spécification de la numérotation de base. Les messages doivent être codés pour la transmission en appliquant les règles de codage de base spécifiées dans la Rec. UIT-T X.690: *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*

A.1 Codage des caractères génériques

L'utilisation des caractères génériques ALL et CHOOSE est autorisée dans le protocole. Cela permet à un contrôleur MGC de spécifier partiellement des identificateurs de terminaison et à une passerelle MG d'effectuer un choix parmi les valeurs qui sont conformes à la spécification partielle. Les identificateurs de terminaison peuvent coder une hiérarchie de noms. Cette hiérarchie est configurée. Par exemple, un identificateur de terminaison peut se composer d'un faisceau de circuits, d'une jonction locale dans ce faisceau et d'un circuit dans cette jonction. Le remplacement par des caractères génériques doit toujours être possible à tous les niveaux. Les alinéas suivants donnent les explications correspondantes.

La description en notation ASN.1 utilise des chaînes d'octets d'une longueur maximale de 8 octets pour les identificateurs de terminaison. C'est-à-dire qu'un identificateur de terminaison se compose d'au plus 64 bits. Un identificateur de terminaison complètement spécifié peut être précédé d'une séquence de champs de remplacement générique, d'une longueur égale à 1 octet. Le bit 7 (de plus fort poids) de cet octet spécifie le type de remplacement générique qui est invoqué: si la valeur du bit est 1, c'est le caractère générique ALL qui est utilisé; si la valeur du bit est 0, c'est le caractère générique CHOOSE qui est utilisé. Le bit 6 du champ de remplacement générique spécifie si le caractère générique appartient à un seul niveau du système hiérarchique de nommage (valeur binaire 0) ou au niveau hiérarchique spécifié dans le champ de remplacement générique plus tous les niveaux inférieurs (valeur binaire 1). Les bits 0 à 5 du champ de remplacement générique spécifient la position binaire dans l'identificateur de terminaison à laquelle commence le remplacement générique.

Quelques exemples illustreront ce modèle. Dans ces exemples, le bit de plus fort poids dans une chaîne de bits apparaît à gauche.

Supposons que les identificateurs de terminaison aient une longueur de trois octets et que chaque octet représente un niveau dans un système hiérarchique de nommage. Un identificateur de terminaison valide sera:

00000001 00011110 01010101.

L'adressage de tous les noms avec le préfixe 00000001 00011110 sera effectué comme suit:

champ de remplacement générique: 10000111

identificateur de terminaison: 00000001 00011110 xxxxxxxx.

Les valeurs des bits représentés par la lettre "x" ne sont pas applicables et doivent être ignorées par le récepteur.

L'indication au récepteur qu'il doit choisir un nom dont le deuxième octet est 00011110 sera effectuée comme suit:

champs de remplacement générique: 00010111 suivi de 00000111

identificateur de terminaison: xxxxxxxx 00011110 xxxxxxxx.

Le premier champ de remplacement générique indique un caractère CHOOSE pour le niveau hiérarchique de nommage commençant au bit 23, qui est le niveau le plus élevé dans notre système de nommage théorique. Le deuxième champ de remplacement générique indique un caractère CHOOSE pour le niveau hiérarchique de nommage commençant au bit 7, qui est le niveau le moins élevé dans notre système de nommage théorique.

Finalement, un nom remplacé par le caractère générique CHOOSE, de niveau hiérarchique de nommage le plus élevé égal à 000000001 est spécifié comme suit:

champ de remplacement générique: 01001111

identificateur de terminaison: 0000001 xxxxxxxx xxxxxxxx.

La valeur binaire 1 à la position binaire 6 du premier octet du champ de remplacement générique indique que le remplacement correspond au niveau spécifié dans la hiérarchie de nommage et à tous les niveaux inférieurs.

Les identificateurs de contexte peuvent également être remplacés par des caractères génériques. Dans ce cas toutefois, la spécification de noms partiels n'est pas autorisée. L'identificateur Context ID 0x0 DOIT être utilisé pour indiquer le contexte néant, l'identificateur Context ID 0xFFFFFFFE DOIT être utilisé pour indiquer un caractère générique CHOOSE et l'identificateur Context ID 0xFFFFFFFF DOIT être utilisé pour indiquer un caractère générique ALL.

L'identificateur de terminaison 0xFFFFFFFFFFFFFFFF DOIT être utilisé pour indiquer la terminaison racine.

A.2 Spécification de syntaxe en notation ASN.1

La spécification en notation ASN.1 de la syntaxe du protocole H.248 est donnée dans le paragraphe ci-dessous.

NOTE 1 – Si l'on utilise un mécanisme de transport faisant appel au verrouillage des niveaux d'application, la définition ci-dessous de l'élément `Transaction` change. Se reporter à l'annexe ou à la Recommandation des séries H.248.x définissant le mécanisme de transport pour la définition applicable à ce cas.

NOTE 2 – La spécification ASN.1 ci-après contient une clause définissant la liste `TerminationIDList` comme étant une séquence d'identificateurs de terminaison (TerminationID). La longueur de cette séquence DOIT être égale à un, sauf, éventuellement, lorsqu'elle est utilisée dans le paramètre contextAuditResult.

NOTE 3 – La présente spécification de syntaxe ne permet pas de faire respecter toutes les restrictions relatives aux inclusions et aux valeurs des éléments. Certaines restrictions supplémentaires sont mentionnées dans des commentaires, tandis que d'autres restrictions apparaissent dans le texte de la présente Recommandation. Ces restrictions supplémentaires font partie du protocole même si leur respect n'est pas assuré par la présente Recommandation.

NOTE 4 – Le module en notation ASN.1 de la présente annexe utilise des types de chaîne d'octets pour le codage des valeurs des paramètres de propriété, de signaux et d'événements et des statistiques. Les types effectifs de ces valeurs varient et sont spécifiés à l'Annexe C ou dans la définition de paquetage applicable.

Une valeur fait d'abord l'objet d'un codage BER en fonction de son type, au moyen du tableau ci-après. Le résultat de ce codage est ensuite codé sous la forme d'une chaîne d'octets en notation ASN.1, en effectuant un "double bouclage" de la valeur. Le format spécifié à l'Annexe C ou le paquetage est lié au codage BER comme indiqué dans le tableau ci-après:

Type spécifié dans le paquetage	Type BER ASN.1
Chaîne	IA5String ou UTF8String (Note 4)
Entier (4 octets)	INTEGER
Double (entier avec signe à 8 octets)	INTEGER (Note 3)
Caractère (UTF-8) (Note 1)	IA5String
Enumération	ENUMERATED
Booléen	BOOLEAN
Entier sans signe (Note 2)	INTEGER (Note 3)
Octet (chaîne)	OCTET STRING
NOTE 1 – Peut avoir plus d'un octet.	
NOTE 2 – L'entier sans signe est référencé à l'Annexe C.	
NOTE 3 – Le codage BER de INTEGER n'implique pas l'emploi de 4 octets.	
NOTE 4 – String devrait être codé comme IA5String lorsque la chaîne ne contient que des caractères ASCII , mais comme UTF8String si elle contient un quelconque caractère non ASCII.	

Voir le § 8.7/X.690 en ce qui concerne la définition du codage d'une valeur de chaîne d'octets.

```
MEDIA-GATEWAY-CONTROL {itu-t(0) recommendation(0) h(8) h248(248) modules(0)
media-gateway-control(0) version2(2)}
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
MegacoMessage ::= SEQUENCE
{
     authHeader         AuthenticationHeader OPTIONAL,
     mess               Message
}

AuthenticationHeader ::= SEQUENCE
{
     secParmIndex       SecurityParmIndex,
     seqNum             SequenceNum,
     ad                 AuthData
}

SecurityParmIndex ::= OCTET STRING(SIZE(4))
SequenceNum          ::= OCTET STRING(SIZE(4))
AuthData                  ::= OCTET STRING (SIZE (12..32))
Message ::= SEQUENCE
{
     version                INTEGER(0..99),
     -- La version du protocole définie ici est égale à 2.
     mId                MId, -- Nom/adresse de l'expéditeur du message
     messageBody CHOICE
     {
       messageError     ErrorDescriptor,
       transactions          SEQUENCE OF Transaction
     },
     ...
}

MId ::= CHOICE
{
```

```
     ip4Address              IP4Address,
     ip6Address              IP6Address,
     domainName              DomainName,
     deviceName              PathName,
     mtpAddress              OCTET STRING(SIZE(2..4)),
      --  Structure d'adressage de mtpAddress:
      --     25 - 15           0
      --        |  PC         | NI |
      --      24 - 14 bits    2 bits
      --  NOTE: 14 bits sont destinés à l'usage international.
      -- Lorsque le code de point est de 16 ou 24 bits, il existe deux options
         nationales.
      -- Afin d'aligner les octets de mtpAddress, les bits de poids le plus fort
      -- seront codés comme étant des zéros
          ...
}

DomainName ::= SEQUENCE
{
     name    IA5String,
     -- Le nom commence par un chiffre alphanumérique suivi d'une
     -- séquence de chiffres alphanumériques, de traits d'union et de points.
     -- Deux points ne peuvent pas être consécutifs.
     portNumber              INTEGER(0..65535) OPTIONAL
}

IP4Address ::= SEQUENCE
{
     address                 OCTET STRING (SIZE(4)),
     portNumber              INTEGER(0..65535) OPTIONAL
}

IP6Address ::= SEQUENCE
{
     address                 OCTET STRING (SIZE(16)),
     portNumber              INTEGER(0..65535) OPTIONAL
}

PathName ::= IA5String(SIZE (1..64))
-- Se reporter au § A.3

Transaction ::= CHOICE
{
     transactionRequest      TransactionRequest,
     transactionPending      TransactionPending,
     transactionReply        TransactionReply,
     transactionResponseAck  TransactionResponseAck,
               -- l'utilisation des acquittements de réponse dépend du transport
                  sous-jacent
     ...
}

TransactionId ::= INTEGER(0..4294967295)       -- entier non signé de 32 bits

TransactionRequest ::= SEQUENCE
{
     transactionId           TransactionId,
     actions                 SEQUENCE OF ActionRequest,
     ...
}

TransactionPending ::= SEQUENCE
{
```

```
     transactionId          TransactionId,
     ...
}

TransactionReply ::= SEQUENCE
{
     transactionId          TransactionId,
     immAckRequired         NULL OPTIONAL,
     transactionResult      CHOICE
     {
          transactionError  ErrorDescriptor,
          actionReplies     SEQUENCE OF ActionReply
     },
     ...
}

TransactionResponseAck ::= SEQUENCE OF TransactionAck
TransactionAck ::= SEQUENCE
{
     firstAck           TransactionId,
     lastAck            TransactionId OPTIONAL
}

ErrorDescriptor ::= SEQUENCE
{
     errorCode          ErrorCode,
     errorText          ErrorText OPTIONAL
}

ErrorCode ::= INTEGER(0..65535)
-- Se reporter au § 13 pour les considérations IANA concernant les codes
    d'erreur

ErrorText ::= IA5String

ContextID ::= INTEGER(0..4294967295)

-- Valeur de contexte néant: 0
-- Valeur de contexte CHOOSE: 4294967294 (0xFFFFFFFE)
-- Valeur de contexte ALL: 4294967295 (0xFFFFFFFF)


ActionRequest ::= SEQUENCE
{
     contextId              ContextID,
     contextRequest         ContextRequest OPTIONAL,
     contextAttrAuditReq    ContextAttrAuditRequest OPTIONAL,
     commandRequests        SEQUENCE OF CommandRequest
}

ActionReply ::= SEQUENCE
{
     contextId          ContextID,
     errorDescriptor    ErrorDescriptor OPTIONAL,
     contextReply       ContextRequest OPTIONAL,
     commandReply       SEQUENCE OF CommandReply
}

ContextRequest ::= SEQUENCE
{
     priority           INTEGER(0..15) OPTIONAL,
     emergency          BOOLEAN OPTIONAL,
```

```
    topologyReq         SEQUENCE OF TopologyRequest OPTIONAL,
    ...
}

ContextAttrAuditRequest ::= SEQUENCE
{
    topology NULL OPTIONAL,
    emergency NULL OPTIONAL,
    priority NULL OPTIONAL,
    ...
}

CommandRequest ::= SEQUENCE
{
    command             Command,
    optional            NULL OPTIONAL,
    wildcardReturn      NULL OPTIONAL,
    ...
}

Command ::= CHOICE
{
    addReq                  AmmRequest,
    moveReq                 AmmRequest,
    modReq                  AmmRequest,
    -- Les demandes Add, Move et Modify ont les mêmes paramètres
    subtractReq             SubtractRequest,
    auditCapRequest         AuditRequest,
    auditValueRequest       AuditRequest,
    notifyReq               NotifyRequest,
    serviceChangeReq        ServiceChangeRequest,
    ...
}

CommandReply ::= CHOICE
{
    addReply                    AmmsReply,
    moveReply                   AmmsReply,
    modReply                    AmmsReply,
    subtractReply               AmmsReply,
    -- Les réponses Add, Move, Modify et Subtract ont les mêmes paramètres
    auditCapReply               AuditReply,
    auditValueReply             AuditReply,
    notifyReply                 NotifyReply,
    serviceChangeReply      ServiceChangeReply,
    ...
}
TopologyRequest ::= SEQUENCE
{
    terminationFrom             TerminationID,
    terminationTo               TerminationID,
    topologyDirection           ENUMERATED
    {
      bothway(0),
      isolate(1),
      oneway(2)
    },
    ...,
    streamID                    StreamID OPTIONAL
}

AmmRequest ::= SEQUENCE
{
    terminationID           TerminationIDList,
```

```
     descriptors             SEQUENCE OF AmmDescriptor,
     -- Au moins un descripteur de chaque type (voir AmmDescriptor)
     -- autorisé dans la séquence.
     ...
}

AmmDescriptor ::= CHOICE
{
     mediaDescriptor              MediaDescriptor,
     modemDescriptor              ModemDescriptor,
     muxDescriptor                MuxDescriptor,
     eventsDescriptor             EventsDescriptor,
     eventBufferDescriptor        EventBufferDescriptor,
     signalsDescriptor            SignalsDescriptor,
     digitMapDescriptor           DigitMapDescriptor,
     auditDescriptor              AuditDescriptor,
...
}


AmmsReply ::= SEQUENCE
{
     terminationID           TerminationIDList,
     terminationAudit        TerminationAudit OPTIONAL,
     ...
}

SubtractRequest ::= SEQUENCE
{
     terminationID           TerminationIDList,
     auditDescriptor         AuditDescriptor OPTIONAL,
     ...
}

AuditRequest ::= SEQUENCE
{
     terminationID           TerminationID,
     auditDescriptor         AuditDescriptor,
     ...
}

AuditReply ::= CHOICE
{
     contextAuditResult      TerminationIDList,
     error     `             ErrorDescriptor,
     auditResult             AuditResult,
     ...
}
AuditResult ::= SEQUENCE
{

     terminationID           TerminationID,
     terminationAuditResult  TerminationAudit
}



TerminationAudit ::= SEQUENCE OF AuditReturnParameter

AuditReturnParameter ::= CHOICE
{
     errorDescriptor              ErrorDescriptor,
     mediaDescriptor              MediaDescriptor,
     modemDescriptor              ModemDescriptor,
```

```
     muxDescriptor               MuxDescriptor,
     eventsDescriptor            EventsDescriptor,
     eventBufferDescriptor       EventBufferDescriptor,
     signalsDescriptor           SignalsDescriptor,
     digitMapDescriptor          DigitMapDescriptor,
     observedEventsDescriptor    ObservedEventsDescriptor,
     statisticsDescriptor        StatisticsDescriptor,
     packagesDescriptor          PackagesDescriptor,
     emptyDescriptors            AuditDescriptor,
...
}

AuditDescriptor ::= SEQUENCE
{
     auditToken  BIT STRING
          {
                muxToken(0), modemToken(1), mediaToken(2),
                eventsToken(3), signalsToken(4),
                digitMapToken(5), statsToken(6),
                observedEventsToken(7),
                packagesToken(8), eventBufferToken(9)
          } OPTIONAL,
     ...,
     auditPropertyToken SEQUENCE OF IndAuditParameter OPTIONAL
}

IndAuditParameter ::= CHOICE
{
     indaudmediaDescriptor          IndAudMediaDescriptor,
     indaudeventsDescriptor         IndAudEventsDescriptor,
     indaudeventBufferDescriptor    IndAudEventBufferDescriptor,
     indaudsignalsDescriptor        IndAudSignalsDescriptor,
     indauddigitMapDescriptor       IndAudDigitMapDescriptor,
     indaudstatisticsDescriptor     IndAudStatisticsDescriptor,
     indaudpackagesDescriptor       IndAudPackagesDescriptor,
     ...
}

IndAudMediaDescriptor ::= SEQUENCE
{

     termStateDescr     IndAudTerminationStateDescriptor OPTIONAL,
     streams     CHOICE
                 {
                 oneStream     IndAudStreamParms,
                 multiStream   SEQUENCE OF IndAudStreamDescriptor
                 } OPTIONAL,
     ...
}

IndAudStreamDescriptor ::= SEQUENCE
{
     streamID                        StreamID,
     streamParms                     IndAudStreamParms
}

IndAudStreamParms ::= SEQUENCE
{
     localControlDescriptor     IndAudLocalControlDescriptor OPTIONAL,
     localDescriptor            IndAudLocalRemoteDescriptor OPTIONAL,
     remoteDescriptor           IndAudLocalRemoteDescriptor OPTIONAL,
     ...
}
```

```
IndAudLocalControlDescriptor ::= SEQUENCE
{
     streamMode          NULL OPTIONAL,
     reserveValue        NULL OPTIONAL,
     reserveGroup        NULL OPTIONAL,
     propertyParms       SEQUENCE OF IndAudPropertyParm OPTIONAL,
     ...
}

IndAudPropertyParm ::= SEQUENCE
{
     name           PkgdName,
     ...
}

IndAudLocalRemoteDescriptor ::= SEQUENCE
{
     propGroupID    INTEGER(0..65535) OPTIONAL,
     propGrps       IndAudPropertyGroup,
     ...
}

IndAudPropertyGroup ::= SEQUENCE OF IndAudPropertyParm

IndAudTerminationStateDescriptor ::= SEQUENCE
{
     propertyParms           SEQUENCE OF IndAudPropertyParm,
     eventBufferControl      NULL OPTIONAL,
     serviceState            NULL OPTIONAL,
     ...
}

IndAudEventsDescriptor ::= SEQUENCE
{
     requestID           RequestID OPTIONAL,
     pkgdName            PkgdName,
     streamID            StreamID OPTIONAL,
     ...
}

IndAudEventBufferDescriptor ::= SEQUENCE
{
     eventName           PkgdName,
     streamID            StreamID OPTIONAL,
     ...
}

IndAudSignalsDescriptor ::=CHOICE
{
     signal              IndAudSignal,
     seqSigList          IndAudSeqSigList,
     ...
}

IndAudSeqSigList ::= SEQUENCE
{
     id          INTEGER(0..65535),
     signalList  IndAudSignal OPTIONAL
}

IndAudSignal ::= SEQUENCE
{
     signalName          PkgdName,
     streamID            StreamID OPTIONAL,
```

```
    ...
}

IndAudDigitMapDescriptor ::= SEQUENCE
{
    digitMapName       DigitMapName       OPTIONAL
}

IndAudStatisticsDescriptor ::= SEQUENCE
{
    statName           PkgdName
}

IndAudPackagesDescriptor ::= SEQUENCE
{
    packageName        Name,
    packageVersion     INTEGER(0..99),
    ...
}
NotifyRequest ::= SEQUENCE
{
    terminationID               TerminationIDList,
    observedEventsDescriptor    ObservedEventsDescriptor,
    errorDescriptor             ErrorDescriptor OPTIONAL,
    ...
}

NotifyReply ::= SEQUENCE
{
    terminationID               TerminationIDList,
    errorDescriptor             ErrorDescriptor OPTIONAL,
    ...
}

ObservedEventsDescriptor ::= SEQUENCE
{
    requestId                   RequestID,
    observedEventLst            SEQUENCE OF ObservedEvent
}
ObservedEvent ::= SEQUENCE
{
    eventName                   EventName,
    streamID                    StreamID OPTIONAL,
    eventParList                SEQUENCE OF EventParameter,
    timeNotation                TimeNotation OPTIONAL,
    ...
}

EventName ::= PkgdName

EventParameter ::= SEQUENCE
{
    eventParameterName          Name,
    value                       Value,
    -- En ce qui concerne l'emploi de extraInfo. voir le commentaire qui porte
       sur PropertyParm
        extraInfo CHOICE
        {
              relation    Relation,
              range       BOOLEAN,
              sublist     BOOLEAN
        } OPTIONAL,
    ...
```

```
}

ServiceChangeRequest ::= SEQUENCE
{
     terminationID               TerminationIDList,
     serviceChangeParms          ServiceChangeParm,
     ...
}

ServiceChangeReply ::= SEQUENCE
{
     terminationID               TerminationIDList,
     serviceChangeResult         ServiceChangeResult,
     ...
}

-- Aucun paramètre n'est obligatoire pour ServiceChangeResult, d'où la
   distinction entre
-- ServiceChangeParm et ServiceChangeResParm.

ServiceChangeResult ::= CHOICE
{
     errorDescriptor             ErrorDescriptor,
     serviceChangeResParms       ServiceChangeResParm
}

WildcardField ::= OCTET STRING(SIZE(1))

TerminationID ::= SEQUENCE
{
     wildcard SEQUENCE OF WildcardField,
     id   OCTET STRING(SIZE(1..8)),
     ...
}
-- Se reporter au § A.1 pour l'explication du mécanisme de remplacement
   générique.
--  L'identificateur  de  terminaison  0xFFFFFFFFFFFFFFFF  désigne  la  terminaison
   racine.

TerminationIDList ::= SEQUENCE OF TerminationID

MediaDescriptor ::= SEQUENCE
{

     termStateDescr TerminationStateDescriptor OPTIONAL,
     streams        CHOICE
                    {
                         oneStream StreamParms,
                         multiStream   SEQUENCE OF StreamDescriptor
                    } OPTIONAL,
     ...
}

StreamDescriptor ::= SEQUENCE
{
     streamID               StreamID,
     streamParms            StreamParms
}

StreamParms ::= SEQUENCE
{
     localControlDescriptor      LocalControlDescriptor OPTIONAL,
     localDescriptor             LocalRemoteDescriptor OPTIONAL,
```

```
    remoteDescriptor            LocalRemoteDescriptor OPTIONAL,
    ...
}

LocalControlDescriptor ::= SEQUENCE
{
    streamMode          StreamMode OPTIONAL,
    reserveValue        BOOLEAN OPTIONAL,
    reserveGroup        BOOLEAN OPTIONAL,
    propertyParms       SEQUENCE OF PropertyParm,
    ...
}

StreamMode ::= ENUMERATED
{
    sendOnly(0),
    recvOnly(1),
    sendRecv(2),
    inactive(3),
    loopBack(4),
      ...
}

-- Dans PropertyParm, la valeur est une chaîne d'octets de type SEQUENCE OF.
-- Lorsqu'elle est envoyée par un contrôleur MGC, l'interprétation est la
-- suivante:
-- une séquence vide signifie CHOOSE,
-- une séquence à 1 seul élément spécifie une valeur,
-- Si le champ sublist n'est pas sélectionné, une séquence plus longue signifie
-- "choisir une des valeurs" (c'est-à-dire valeur1 OU valeur2 OU…)
-- Si le champ sublist est sélectionné,
-- une séquence de plus d'un élément code la valeur
-- d'une propriété à liste de valeur (par exemple valeur1 ET valeur2 ET …).
-- Le champ de relation ne peut être choisi que si la séquence de valeur a la
-- longueur 1. Il indique que la passerelle média doit choisir une valeur pour
-- la propriété. Par exemple, x > 3 indique que la passerelle média doit choisir
-- n'importe quelle valeur supérieure à 3 pour la propriété x.
-- Le champ d'étendue ne peut être choisi que si la séquence de valeur
-- a la longueur 2. Il indique que la passerelle MG doit choisir une valeur
-- dans l'étendue comprise entre le premier et le dernier octet de la
-- séquence, y compris les valeurs des bornes.
-- Lorsqu'elle est envoyée par la passerelle MG, seules les réponses à une
-- demande AuditCapability peuvent contenir plusieurs valeurs, une plage ou
-- un champ de relation.

PropertyParm ::= SEQUENCE
{
    name            PkgdName,
    value           SEQUENCE OF OCTET STRING,
    extraInfo       CHOICE
    {
        relation        Relation,
        range           BOOLEAN,
        sublist         BOOLEAN
    } OPTIONAL,
    ...
}

Name ::= OCTET STRING(SIZE(2))

PkgdName ::= OCTET STRING(SIZE(4))
-- représente un nom de paquetage (2 octets) plus une propriété,
-- un événement, des noms de signal ou un identificateur de
-- statistiques (2 octets)
```

```
-- Pour remplacer génériquement un paquetage, utiliser 0xFFFF
-- pour les deux premiers octets. Le caractère CHOOSE n'est pas autorisé.
-- Pour faire référence à l'étiquette de propriété initiale spécifiée dans
-- l'Annexe C, utiliser 0x0000 dans les deux premiers octets. Le remplacement
-- générique du nom d'un paquetage n'est permis que si une propriété,
-- un événement, un nom de signal ou un identificateur de statistique
-- sont également remplacés génériquement.

Relation ::= ENUMERATED
{
     greaterThan(0),
     smallerThan(1),
     unequalTo(2),
     ...
}

LocalRemoteDescriptor ::= SEQUENCE
{
     propGrps SEQUENCE OF PropertyGroup,
     ...
}

PropertyGroup ::= SEQUENCE OF PropertyParm

TerminationStateDescriptor ::= SEQUENCE
{
     propertyParms          SEQUENCE OF PropertyParm,
     eventBufferControl   EventBufferControl OPTIONAL,
     serviceState           ServiceState OPTIONAL,
     ...
}

EventBufferControl ::= ENUMERATED
{
     off(0),
     lockStep(1),
     ...
}

ServiceState ::= ENUMERATED
{
     test(0),
     outOfSvc(1),
     inSvc(2),
     ...
}

MuxDescriptor   ::= SEQUENCE
{
     muxType            MuxType,
     termList           SEQUENCE OF TerminationID,
     nonStandardData    NonStandardData OPTIONAL,
     ...
}
MuxType ::= ENUMERATED
{
     h221(0),
     h223(1),
     h226(2),
     v76(3),
     ...,
     nx64k(4)
}
```

```
StreamID ::= INTEGER(0..65535)   -- entier de 16 bits non signé

EventsDescriptor ::= SEQUENCE
{
     requestID          RequestID OPTIONAL,
                        -- RequestID doit être présent même si eventList
                        -- est non vide
     eventList          SEQUENCE OF RequestedEvent,
     ...
}

RequestedEvent ::= SEQUENCE
{
     pkgdName           PkgdName,
     streamID           StreamID OPTIONAL,
     eventAction        RequestedActions OPTIONAL,
     evParList          SEQUENCE OF EventParameter,
     ...
}

RequestedActions ::= SEQUENCE
{
     keepActive              BOOLEAN OPTIONAL,
     eventDM                 EventDM OPTIONAL,
     secondEvent             SecondEventsDescriptor OPTIONAL,
     signalsDescriptor       SignalsDescriptor OPTIONAL,
     ...
}


EventDM ::= CHOICE
{    digitMapName  DigitMapName,
     digitMapValue DigitMapValue
}

SecondEventsDescriptor ::= SEQUENCE
{
     requestID               RequestID OPTIONAL,
     eventList               SEQUENCE OF SecondRequestedEvent,
     ...
}

SecondRequestedEvent ::= SEQUENCE
{
     pkgdName                PkgdName,
     streamID                StreamID OPTIONAL,
     eventAction             SecondRequestedActions OPTIONAL,
     evParList               SEQUENCE OF EventParameter,
     ...
}
SecondRequestedActions ::= SEQUENCE
{
     keepActive              BOOLEAN OPTIONAL,
     eventDM                 EventDM OPTIONAL,
     signalsDescriptor       SignalsDescriptor OPTIONAL,
     ...
}

EventBufferDescriptor ::= SEQUENCE OF EventSpec

EventSpec ::= SEQUENCE
{
     eventName   EventName,
     streamID    StreamID OPTIONAL,
```

```
     eventParList       SEQUENCE OF EventParameter,
     ...
}


SignalsDescriptor ::= SEQUENCE OF SignalRequest

SignalRequest ::=CHOICE
{
     signal             Signal,
     seqSigList         SeqSigList,
     ...
}

SeqSigList ::= SEQUENCE
{
     id             INTEGER(0..65535),
     signalList     SEQUENCE OF Signal
}

Signal ::= SEQUENCE
{
     signalName         SignalName,
     streamID       StreamID OPTIONAL,
     sigType        SignalType OPTIONAL,
     duration       INTEGER (0..65535) OPTIONAL,
     notifyCompletion   NotifyCompletion OPTIONAL,
     keepActive         BOOLEAN OPTIONAL,
     sigParList         SEQUENCE OF SigParameter,
     ...
}

SignalType ::= ENUMERATED
{
     brief(0),
     onOff(1),
     timeOut(2),
     ...
}

SignalName ::= PkgdName

NotifyCompletion ::= BIT STRING
{
     onTimeOut(0), onInterruptByEvent(1),
     onInterruptByNewSignalDescr(2), otherReason(3)
}

SigParameter ::= SEQUENCE
{
     sigParameterName       Name,
     value                  Value,
     -- En ce qui concerne l'emploi de extraInfo voir
     -- le commentaire qui porte sur PropertyParm
          extraInfo CHOICE
          {
               relation      Relation,
              range          BOOLEAN,
              sublist        BOOLEAN
          } OPTIONAL,
     ...
}
```

```
-- Pour une AuditCapReply avec tous les événements,
-- le RequestID DOIT être ALL.
-- ALL est représenté par 0xffffffff.

RequestID ::= INTEGER(0..4294967295)   -- entier de 32 bits non signé

ModemDescriptor ::= SEQUENCE
{
     mtl  SEQUENCE OF ModemType,
     mpl  SEQUENCE OF PropertyParm,
     nonStandardData    NonStandardData OPTIONAL
}

ModemType ::= ENUMERATED
{
     v18(0),
     v22(1),
     v22bis(2),
     v32(3),
     v32bis(4),
     v34(5),
     v90(6),
     v91(7),
     synchISDN(8),
     ...
}

DigitMapDescriptor ::= SEQUENCE
{
     digitMapName  DigitMapName  OPTIONAL,
     digitMapValue DigitMapValue OPTIONAL
}

DigitMapName ::= Name

DigitMapValue ::= SEQUENCE
{
     startTimer    INTEGER(0..99) OPTIONAL,
     shortTimer    INTEGER(0..99) OPTIONAL,
     longTimer     INTEGER(0..99) OPTIONAL,
     digitMapBody  IA5String,
          -- Les unités sont exprimées en secondes pour la temporisation de
          -- début et les temporisations courtes et longues, et en centaines de
          -- millisecondes pour la temporisation de durée. Donc les intervalles
          -- de début, les intervalles courts et longs s'étendent de 1 à
          -- 99 secondes, tandis que ceux pour la durée s'étendent de 100 ms
          -- à 9,9 s
          -- Se reporter au § A.3 pour l'explication de la syntaxe d'un sujet de
          -- numérotation
     ...,
     durationTimer      INTEGER (0..99) OPTIONAL
}

ServiceChangeParm ::= SEQUENCE
{
     serviceChangeMethod     ServiceChangeMethod,
     serviceChangeAddress    ServiceChangeAddress OPTIONAL,
     serviceChangeVersion    INTEGER(0..99) OPTIONAL,
     serviceChangeProfile    ServiceChangeProfile OPTIONAL,
     serviceChangeReason     Value,
          -- Un motif serviceChangeReason comporte un code numérique indiquant
          -- la raison et une description alphanumérique facultative.
          -- Le motif serviceChangeReason DOIT être une chaîne comportant un
          -- code décimal indiquant la raison, suivi facultativement d'un
```

```
        -- blanc unique et d'une chaîne alphanumérique descriptive.
        -- Cette chaîne fait d'abord l'objet d'un codage BER de
        -- forme IA5String.
        -- Le résultat de ce codage BER est ensuite codé comme étant de type
        -- ASN.1 OCTET STRING, à "double bouclage" de la valeur comme pour
        -- les éléments de paquetage.
    serviceChangeDelay      INTEGER(0..4294967295) OPTIONAL,
                                        -- entier de 32 bits non signé
    serviceChangeMgcId      MId OPTIONAL,
    timeStamp               TimeNotation OPTIONAL,
    nonStandardData         NonStandardData OPTIONAL,
    ...,

    serviceChangeInfo  AuditDescriptor OPTIONAL
}

ServiceChangeAddress ::= CHOICE
{
    portNumber              INTEGER(0..65535),    -- numéro d'accès TCP/UDP
    ip4Address              IP4Address,
    ip6Address              IP6Address,
    domainName              DomainName,
    deviceName              PathName,
    mtpAddress              OCTET STRING(SIZE(2..4)),
    ...
}

ServiceChangeResParm ::= SEQUENCE
{
    serviceChangeMgcId      MId OPTIONAL,
    serviceChangeAddress    ServiceChangeAddress OPTIONAL,
    serviceChangeVersion    INTEGER(0..99) OPTIONAL,
    serviceChangeProfile    ServiceChangeProfile OPTIONAL,
    timestamp               TimeNotation OPTIONAL,
...
}

ServiceChangeMethod ::= ENUMERATED
{
    failover(0),
    forced(1),
    graceful(2),
    restart(3),
    disconnected(4),
    handOff(5),
    ...
}

ServiceChangeProfile ::= SEQUENCE
{
    profileName        IA5String(SIZE (1..67))

    -- 64 caractères pour le nom, 1 pour "/", 2 pour la version correspondant
    -- au formulaire ABNF
}

PackagesDescriptor ::= SEQUENCE OF PackagesItem

PackagesItem ::= SEQUENCE
{
    packageName        Name,
    packageVersion     INTEGER(0..99),
    ...
}
```

```
StatisticsDescriptor ::= SEQUENCE OF StatisticsParameter

StatisticsParameter ::= SEQUENCE
{
     statName           PkgdName,
     statValue          Value OPTIONAL
}

NonStandardData ::= SEQUENCE
{
     nonStandardIdentifier   NonStandardIdentifier,
     data                    OCTET STRING
}

NonStandardIdentifier ::= CHOICE
{
     object                  OBJECT IDENTIFIER,
     h221NonStandard         H221NonStandard,
     experimental            IA5String(SIZE(8)),
           -- les deux premiers caractères doivent être "X–" ou "X+"
     ...
}

H221NonStandard ::= SEQUENCE
{    t35CountryCode1         INTEGER(0..255),
     t35CountryCode2         INTEGER(0..255),     -- pays, conformément à T.35
     t35Extension            INTEGER(0..255),     -- attribué nationalement
     manufacturerCode        INTEGER(0..65535),  -- attribué nationalement
     ...
}

TimeNotation ::= SEQUENCE
{
     date     IA5String(SIZE(8)),      -- format yyyymmdd
     time     IA5String(SIZE(8))       -- format hhmmssss
     -- conformément à l'ISO 8601:1988
}

Value ::= SEQUENCE OF OCTET STRING


END
```

A.3 Noms des scripts de numérotation et des conduits

Du point de vue syntaxique, les scripts de numérotation sont des chaînes soumises à des contraintes de syntaxe. La syntaxe des scripts de numérotation valides est spécifiée en formalisme ABNF (RFC 2234). La syntaxe de script de numérotation présentée dans le présent paragraphe n'a valeur que d'illustration. La définition de la structure `digitmap` dans l'Annexe B a priorité en cas de divergence entre les deux définitions.

```
digitMap = (digitString / LWSP "(" LWSP digitStringList LWSP ")" LWSP)

digitStringList = digitString *( LWSP "|" LWSP digitString )

digitString = 1*(digitStringElement)

digitStringElement = digitPosition [DOT]

digitPosition = digitMapLetter / digitMapRange

digitMapRange = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))

digitLetter = *((DIGIT "-" DIGIT) /digitMapLetter)
```

```
digitMapLetter = DIGIT              ;chiffres 0 à 9
              / %x41-4B / %x61-6B    ;de "a" à "k" et de "A" à "K"
              / "L"     / "S" /     "T"   ;Temporisateurs interévénements
                                            ;(longs, courts, start)
              / "Z"                         ;événement de longue durée
DOT = %x2E ; "."
LWSP = *(WSP / COMMENT / EOL)
WSP = SP / HTAB
COMMENT = ";" *(SafeChar / RestChar / WSP) EOL
EOL = (CR [LF]) / LF
SP = %x20
HTAB = %x09
CR = %x0D
LF = %x0A
SafeChar = DIGIT / ALPHA / "+" / "-" / "&" / "!" / "_" / "/" /
          "'" / "?" / "@" / "^" / "`" / "~" / "*" / "$" / "\" /
          "(" / ")" / "%" / "."
RestChar = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
          "<" / ">" / "=" / %x22
DIGIT = %x30-39         ; digits 0 through 9
ALPHA = %x41-5A / %x61-7A     ; A-Z, a-z
```

Un nom de conduit est également une chaîne soumise à des contraintes syntaxiques. La production ABNF qui définit cette chaîne est reprise de l'Annexe B.

```
; La longueur totale de pathNAME ne doit pas dépasser 64 caractères.
pathNAME             = ["*"] NAME *("/" / "*"/ ALPHA / DIGIT /"_" / "$" )
                       ["@" pathDomainName ]

; ABNF admet consécutivement deux ou plusieurs symboles "." même si cela n'a
; pas de sens
; dans un nom de domaine de conduits.
pathDomainName       = (ALPHA / DIGIT / "*" )
                       *63(ALPHA / DIGIT / "-" / "*" / ".")
NAME = ALPHA *63(ALPHA / DIGIT / "_" )
```

Annexe B

Codage alphanumérique du protocole

B.1 Codage des caractères génériques

Lors du codage alphanumérique du protocole, les identificateurs de terminaison sont certes arbitraires mais un choix judicieux des noms permet de rendre plus utile le caractère générique "*". Lorsque ce caractère générique est rencontré, il "correspond" à tous les identificateurs de terminaison dont les caractères précédents et suivants sont les mêmes (le cas échéant). Par exemple, en présence des identificateurs de terminaison R13/3/1, R13/3/2 et R13/3/3, l'identificateur de terminaison R13/3/* leur correspondra tous. Dans certains cas, toutes les terminaisons doivent être désignées. L'identificateur de terminaison "*" suffit car il correspond au caractère ALL. Le caractère "$" peut être utilisé comme caractère "CHOOSE" dans un identificateur de terminaison afin de signaler à la passerelle MG qu'elle doit créer une terminaison éphémère ou choisir une terminaison physique au repos.

B.2 Spécification en formalisme ABNF

La syntaxe du protocole est présentée en formalisme ABNF conformément au commentaire RFC 2234.

NOTE 1 – La présente spécification de syntaxe ne permet pas de faire respecter toutes les restrictions relatives aux inclusions et aux valeurs des éléments. Certaines restrictions supplémentaires sont mentionnées dans des commentaires, tandis que d'autres restrictions apparaissent dans le texte de la présente Recommandation. Ces restrictions supplémentaires font partie du protocole même si leur respect n'est pas assuré par la présente Recommandation.

NOTE 2 – La syntaxe dépend du contexte. Par exemple, "Add" peut être le jeton AddToken ou un nom NAME, en fonction du contexte dans lequel il figure.

Le formalisme ABNF et le codage alphanumérique sont insensibles à la casse. Cela inclut les identificateurs de terminaison, de script de scénario, etc. Le protocole SDP est sensible à la casse en vertu de RFC 2327.

```
; NOTE -- Le formalisme ABNF de la présente section emploie la structure VALUE
; (ou des listes de structures VALUE) pour le codage des diverses valeurs
; d'élément de paquetage (propriétés, paramètres des signaux, etc.). Les types
; de ces valeurs varient et sont spécifiés dans la définition de paquetage
; applicable. Plusieurs de ces types sont décrits au § 12.2.
;
; La spécification du formalisme ABNF pour VALUE admet une forme quotedString ou
; un ensemble de SafeChars. Le codage des valeurs d'élément de paquetage en
; VALUES ABNF est spécifié ci-après. Si le codage d'un type admet des
; caractères autres que SafeChars, la forme quotedString DOIT être employée pour
; toutes les valeurs de ce type, même pour des valeurs spécifiques qui ne
; comportent que des SafeChars.
;
; Chaîne: une chaîne DOIT employer la forme quotedString de VALUE et peut
; contenir tout caractère admis dans cette forme quotedString.
;
; Entier, double et sans signe: les valeurs décimales peuvent être codées au
; moyen des caractères 0 à 9. Les valeurs hexadécimales doivent être précédées
; de '0x' et peuvent utiliser les caractères 0 à 9, a à f et A à F. Le format
; octal n'est pas admis. Les entiers négatifs commençant pas le signe '-'
; DOIVENT être décimaux. La forme SafeChar de VALUE DOIT être employée.
;
; Caractère: un codage UTF-8 d'une seule lettre précédée et suivie de
; guillemets.
;
```

```
; Enumération: une énumération DOIT employer la forme SafeChar de VALUE et peut
; contenir tout caractère admis dans cette forme SafeChar.
;
; Booléen: les valeurs booléennes sont codées comme "on" et "off" et ne sont pas
; sensibles à la casse. La forme SafeChar de VALUE DOIT être utilisée.
;
; Types ultérieurs: tout type défini DOIT satisfaire à la spécification ABNF de
; VALUE. En particulier, si le codage d'un type admet des caractères autres que
; SafeChars, la forme quotedString DOIT être employée pour toutes les valeurs de
; ce type, même pour des valeurs spécifiques qui ne comportent que des
; SafeChars.
;
; Il convient de noter qu'il n'y a pas moyen d'employer les guillemets dans une
; valeur.
;
; Il convient aussi de noter que le protocole SDP n'admet pas les blancs en
; début de ligne, tandis que le formalisme ABNF du contrôleur MGC autorise les
; blancs avant que le protocole SDP ne débute au niveau du descripteur local ou
; distant. Les analyseurs syntaxiques devraient accepter les blancs entre le
; LBRKT suivant le jeton local ou distant et le début du protocole SDP.

megacoMessage        = LWSP [authenticationHeader SEP ] message

authenticationHeader = AuthToken EQUAL SecurityParmIndex COLON
                       SequenceNum COLON AuthData

SecurityParmIndex    = "0x" 8(HEXDIG)
SequenceNum          = "0x" 8(HEXDIG)
AuthData             = "0x" 24*64(HEXDIG)

message            = MegacopToken SLASH Version SEP mId SEP messageBody
; La version du protocole est définie ici comme étant égale à 2.

messageBody          = ( errorDescriptor / transactionList )

transactionList      = 1*( transactionRequest / transactionReply /
                       transactionPending / transactionResponseAck )
; L'utilisation des acquittements de réponse dépend du transport sous-jacent

transactionPending   = PendingToken EQUAL TransactionID LBRKT RBRKT

transactionResponseAck = ResponseAckToken LBRKT transactionAck
             *(COMMA transactionAck) RBRKT
transactionAck = TransactionID / (TransactionID "-" TransactionID)

transactionRequest   = TransToken EQUAL TransactionID LBRKT
                       actionRequest *(COMMA actionRequest) RBRKT

actionRequest        = CtxToken EQUAL ContextID LBRKT ((
                       contextRequest [COMMA  commandRequestList])
                       / commandRequestList) RBRKT

contextRequest   = ((contextProperties [COMMA contextAudit])
            / contextAudit)

contextProperties    = contextProperty *(COMMA contextProperty)

; une fois au plus
contextProperty      = (topologyDescriptor / priority / EmergencyToken)

contextAudit    = ContextAuditToken LBRKT
                       contextAuditProperties *(COMMA
                       contextAuditProperties) RBRKT
```

```
; une fois au plus
contextAuditProperties = ( TopologyToken / EmergencyToken /
                           PriorityToken )

; "O-" indique une commande facultative
; "W-" indique une réponse générique à une commande
commandRequestList= ["O-"] ["W-"] commandRequest *
(COMMA ["O-"] ["W-"]commandRequest)

commandRequest       = ( ammRequest / subtractRequest / auditRequest /
                         notifyRequest / serviceChangeRequest)

transactionReply     = ReplyToken EQUAL TransactionID LBRKT
                     [ ImmAckRequiredToken COMMA]
               ( errorDescriptor / actionReplyList ) RBRKT

actionReplyList      = actionReply *(COMMA actionReply )

actionReply          = CtxToken EQUAL ContextID LBRKT
                       ( errorDescriptor / commandReply /
                       (commandReply COMMA errorDescriptor) ) RBRKT

commandReply       = (( contextProperties [COMMA commandReplyList] ) /
                         commandReplyList )


commandReplyList     = commandReplys *(COMMA commandReplys )

commandReplys        = (serviceChangeReply / auditReply / ammsReply /
                         notifyReply )

;Add Move et Modify ont les même paramètres de demande
ammRequest           = (AddToken / MoveToken / ModifyToken ) EQUAL
                       TerminationID [LBRKT ammParameter *(COMMA
                       ammParameter) RBRKT]

;une fois au plus
ammParameter         = (mediaDescriptor / modemDescriptor /
                         muxDescriptor / eventsDescriptor /
                         signalsDescriptor / digitMapDescriptor /
                         eventBufferDescriptor / auditDescriptor)

ammsReply            = (AddToken / MoveToken / ModifyToken /
                         SubtractToken ) EQUAL TerminationID [ LBRKT
                         terminationAudit RBRKT ]

subtractRequest      =  SubtractToken EQUAL TerminationID
                         [ LBRKT auditDescriptor RBRKT]

auditRequest         =  (AuditValueToken / AuditCapToken ) EQUAL
                         TerminationID LBRKT auditDescriptor RBRKT

auditReply           = (AuditValueToken / AuditCapToken )
                       ( contextTerminationAudit  / auditOther)

auditOther           = EQUAL TerminationID [LBRKT
                       terminationAudit RBRKT]

terminationAudit   = auditReturnParameter *(COMMA auditReturnParameter)

contextTerminationAudit = EQUAL CtxToken ( terminationIDList /
                        LBRKT errorDescriptor RBRKT )
```

```
auditReturnParameter = (mediaDescriptor / modemDescriptor /
                        muxDescriptor / eventsDescriptor /
                        signalsDescriptor / digitMapDescriptor /
                       observedEventsDescriptor / eventBufferDescriptor /
                        statisticsDescriptor / packagesDescriptor /
                         errorDescriptor / auditItem)

auditDescriptor      = AuditToken LBRKT [ auditItem
                       *(COMMA auditItem) ] RBRKT

notifyRequest        = NotifyToken EQUAL TerminationID
                       LBRKT ( observedEventsDescriptor
                             [ COMMA errorDescriptor ] ) RBRKT

notifyReply          = NotifyToken EQUAL TerminationID
                       [ LBRKT errorDescriptor RBRKT ]

serviceChangeRequest = ServiceChangeToken EQUAL TerminationID
                       LBRKT serviceChangeDescriptor RBRKT

serviceChangeReply   = ServiceChangeToken EQUAL TerminationID
                       [LBRKT (errorDescriptor /
                       serviceChangeReplyDescriptor) RBRKT]

errorDescriptor   = ErrorToken EQUAL ErrorCode
                    LBRKT [quotedString] RBRKT

ErrorCode            = 1*4(DIGIT) ; could be extended

TransactionID        = UINT32

mId                  = (( domainAddress / domainName )
                       [":" portNumber]) / mtpAddress / deviceName

; Le formalisme ABNF autorise deux ou plusieurs "." consécutifs bien que cela
; n'ait pas de sens dans un mom de domaine.
domainName           = "<" (ALPHA / DIGIT) *63(ALPHA / DIGIT / "-" /
                       ".") ">"
deviceName           = pathNAME

;Les valeurs 0x0, 0xFFFFFFFE et 0xFFFFFFFF sont réservées.
ContextID            = (UINT32 / "*" / "-" / "$")

domainAddress        = "[" (IPv4address / IPv6address) "]"
;RFC 2373 contient la définition de IP6Addresses.
IPv6address          = hexpart [ ":" IPv4address ]
IPv4address          = V4hex DOT V4hex DOT V4hex DOT V4hex
V4hex                = 1*3(DIGIT) ; "0".."255"
; cette production, tout en apparaissant dans RFC 2373, n'est pas référencée
; IPv6prefix           = hexpart SLASH 1*2DIGIT
hexpart             = hexseq "::" [ hexseq ] / "::" [ hexseq ] / hexseq
hexseq               = hex4 *( ":" hex4)
hex4                 = 1*4HEXDIG

portNumber           = UINT16

; Structure d'adressage de mtpAddress:
; 25 - 15      0
;    | PC    | NI |
;   24 - 14 bits   2 bits
; Note: 14 bits sont destinés à l'usage international.
; Lorsque le code de point est de 16 ou 24 bits, il existe deux options
; nationales.
```

```
; Afin d'aligner les octets de mtpAddress, les bits de poids le plus fort
; seront codés comme étant des zéros.
; Un octet sera représenté par 2 chiffres hexadécimaux.

mtpAddress          = MTPToken LBRKT 4*8 (HEXDIG) RBRKT

terminationIDList   = LBRKT TerminationID *(COMMA TerminationID) RBRKT

; La longueur totale de pathNAME ne doit pas dépasser 64 caractères.
pathNAME            = ["*"] NAME *("/" / "*"/ ALPHA / DIGIT /"_" / "$" )
                      ["@" pathDomainName ]

; Le formalisme ABNF autorise deux ou plusieurs "." consécutifs bien que cela
; n'ait pas de sens dans un mom de domaine.
pathDomainName      = (ALPHA / DIGIT / "*" )
                      *63(ALPHA / DIGIT / "-" / "*" / ".")

TerminationID       = "ROOT" / pathNAME / "$" / "*"

mediaDescriptor   = MediaToken LBRKT mediaParm *(COMMA mediaParm) RBRKT

; au plus un terminationStateDescriptor
; et soit streamParm(s), soit streamDescriptor(s), mais pas les deux
mediaParm           = (streamParm / streamDescriptor /
                       terminationStateDescriptor)

; une fois au plus par item
streamParm          = ( localDescriptor / remoteDescriptor /
                       localControlDescriptor )

streamDescriptor    = StreamToken EQUAL StreamID LBRKT streamParm
                      *(COMMA streamParm) RBRKT

localControlDescriptor = LocalControlToken LBRKT localParm
                         *(COMMA localParm) RBRKT

; une fois au plus par item sauf pour propertyParm
localParm           = ( streamMode / propertyParm / reservedValueMode
     / reservedGroupMode )

reservedValueMode    = ReservedValueToken EQUAL ( "ON" / "OFF" )
reservedGroupMode    = ReservedGroupToken EQUAL ( "ON" / "OFF" )

streamMode          = ModeToken EQUAL streamModes

streamModes         = (SendonlyToken / RecvonlyToken / SendrecvToken /
                      InactiveToken / LoopbackToken )

propertyParm        = pkgdName parmValue
parmValue           = (EQUAL alternativeValue/ INEQUAL VALUE)
alternativeValue    = ( VALUE
          / LSBRKT VALUE *(COMMA VALUE) RSBRKT
                        ; sublist (i.e. A AND B AND ...)
                / LBRKT VALUE *(COMMA VALUE) RBRKT
               ; alternatives (i.e. A OR B OR ...)

          /  LSBRKT VALUE COLON VALUE RSBRKT )
           ; range


INEQUAL             = LWSP (">" / "<" / "#" ) LWSP
LSBRKT              = LWSP "[" LWSP
RSBRKT              = LWSP "]" LWSP
```

```
; Note – L'octet zéro ne figure pas parmi les caractères autorisés dans la
; chaîne d'octets. Puisque la définition en vigueur se limite au protocole
; SDP, et que l'octet zéro serait un caractère illicite dans ce protocole,
; ceci ne pose pas de problème.
localDescriptor      = LocalToken LBRKT octetString RBRKT

remoteDescriptor     = RemoteToken LBRKT octetString RBRKT

eventBufferDescriptor= EventBufferToken [ LBRKT eventSpec
                       *( COMMA eventSpec) RBRKT ]

eventSpec = pkgdName [ LBRKT eventSpecParameter
              *(COMMA eventSpecParameter) RBRKT ]
eventSpecParameter   = (eventStream / eventOther)

eventBufferControl     = BufferToken EQUAL ( "OFF" / LockStepToken )

terminationStateDescriptor = TerminationStateToken LBRKT
            terminationStateParm *( COMMA terminationStateParm ) RBRKT

; une fois au plus par item sauf pour propertyParm
terminationStateParm =(propertyParm / serviceStates / eventBufferControl )

serviceStates        = ServiceStatesToken EQUAL ( TestToken /
                       OutOfSvcToken / InSvcToken )

muxDescriptor        = MuxToken EQUAL MuxType  terminationIDList

MuxType              = ( H221Token / H223Token / H226Token / V76Token
                         / extensionParameter / Nx64kToken )

StreamID             = UINT16
pkgdName             = (PackageName SLASH ItemID) ;specific item
                       / (PackageName SLASH "*") ;tous les items dans le
                        paquetage
                       / ("*" SLASH "*") ; tous les items pris en charge par la
                        passerelle MG
PackageName          = NAME
ItemID               = NAME

eventsDescriptor     = EventsToken [ EQUAL RequestID LBRKT
                       requestedEvent *( COMMA requestedEvent ) RBRKT ]

requestedEvent       = pkgdName [ LBRKT eventParameter
                       *( COMMA eventParameter ) RBRKT ]

; une fois au plus pour chaque KeepActiveToken , eventDM et eventStream
; au plus un parmi embedWithSig ou embedNoSig mais pas les deux
; KeepActiveToken et embedWithSig ne doivent pas être présents tous les deux
eventParameter       = ( embedWithSig / embedNoSig / KeepActiveToken
                       /eventDM / eventStream / eventOther )

embedWithSig         = EmbedToken LBRKT signalsDescriptor
                       [COMMA embedFirst ] RBRKT
embedNoSig      = EmbedToken LBRKT embedFirst RBRKT

; une fois au plus pour chacun
embedFirst      = EventsToken [ EQUAL RequestID LBRKT
               secondRequestedEvent *(COMMA secondRequestedEvent) RBRKT ]

secondRequestedEvent = pkgdName [ LBRKT secondEventParameter
                       *( COMMA secondEventParameter ) RBRKT ]
```

```
; une fois au plus pour chacun parmi embedSig , KeepActiveToken, eventDM ou
; eventStream
; KeepActiveToken et embedSig ne doivent pas être présents tous les deux.
secondEventParameter = ( embedSig / KeepActiveToken / eventDM /
                         eventStream / eventOther )

embedSig  = EmbedToken LBRKT signalsDescriptor RBRKT

eventStream          = StreamToken EQUAL StreamID

eventOther           = eventParameterName parmValue

eventParameterName   = NAME

eventDM              = DigitMapToken EQUAL(( digitMapName ) /
                       (LBRKT digitMapValue RBRKT ))

signalsDescriptor    = SignalsToken LBRKT [ signalParm
                       *(COMMA signalParm)] RBRKT

signalParm           = signalList / signalRequest

signalRequest        = signalName [ LBRKT sigParameter
                       *(COMMA sigParameter) RBRKT ]

signalList           = SignalListToken EQUAL signalListId LBRKT
                       signalListParm *(COMMA signalListParm) RBRKT

signalListId         = UINT16

; exactement un seul signalType, durée une fois au plus et chaque signal
; parameter
signalListParm       = signalRequest

signalName           = pkgdName
; une fois au plus pour sigStream, une fois au plus pour sigSignalType,
; une fois au plus pour sigDuration, chaque signalParameterName au plus une
; fois.
sigParameter       = sigStream / sigSignalType / sigDuration / sigOther
                   / notifyCompletion / KeepActiveToken
sigStream            = StreamToken EQUAL StreamID
sigOther             = sigParameterName parmValue
sigParameterName     = NAME
sigSignalType        = SignalTypeToken EQUAL signalType
signalType           = (OnOffToken / TimeOutToken / BriefToken)
sigDuration          = DurationToken EQUAL UINT16
notifyCompletion     = NotifyCompletionToken EQUAL (LBRKT
          notificationReason *(COMMA notificationReason) RBRKT)

notificationReason   = ( TimeOutToken / InterruptByEventToken
                       / InterruptByNewSignalsDescrToken
                       / OtherReasonToken )
observedEventsDescriptor = ObservedEventsToken EQUAL RequestID
                   LBRKT observedEvent *(COMMA observedEvent) RBRKT

; temps par événement, car il peut être mémorisé dans le tampon.
observedEvent        = [ TimeStamp LWSP COLON] LWSP
                       pkgdName [ LBRKT observedEventParameter
                       *(COMMA observedEventParameter) RBRKT ]

; une fois au plus pour eventStream, chaque eventParameterName une fois au plus
observedEventParameter = eventStream / eventOther
```

```
; dans le cas où AuditCapReply s'applique à tous les événements, RequestID
; devrait être ALL.
RequestID            = ( UINT32 / "*" )

modemDescriptor      = ModemToken (( EQUAL modemType) /
                       (LSBRKT modemType *(COMMA modemType) RSBRKT))
                       [ LBRKT propertyParm
                        *(COMMA propertyParm) RBRKT ]

; une fois au plus sauf pour extensionParameter
modemType            = (V32bisToken / V22bisToken / V18Token /
                        V22Token / V32Token / V34Token / V90Token /
                        V91Token / SynchISDNToken / extensionParameter)

digitMapDescriptor   = DigitMapToken EQUAL
                       ( ( LBRKT digitMapValue RBRKT )
               / (digitMapName [ LBRKT digitMapValue RBRKT ]) )
digitMapName         = NAME
digitMapValue        = ["T" COLON Timer COMMA] ["S" COLON Timer COMMA]
                       ["L" COLON Timer COMMA] ["Z" COLON Timer COMMA]
                       digitMap
Timer                = 1*2DIGIT
; Les unités sont exprimées en secondes pour les temporisations T, S, et L et en
; centaines de millisecondes pour la temporisation Z. Donc T, S,
; et L s'étendent de 1 à 99 secondes, tandis que les intervalles Z s'étendent de
; 100 ms à 9,9 s.
digitMap = (digitString / LWSP "(" LWSP digitStringList LWSP ")" LWSP)
digitStringList      = digitString *( LWSP "|" LWSP digitString )
digitString          = 1*(digitStringElement)
digitStringElement   = digitPosition [DOT]
digitPosition        = digitMapLetter / digitMapRange
digitMapRange        = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))
digitLetter          = *((DIGIT "-" DIGIT ) / digitMapLetter)
digitMapLetter       = DIGIT   ;Basic event symbols
                      / %x41-4B / %x61-6B ; a-k, A-K
                      / "L" / "S" / "T" ;Inter-event timers
                              ; (long, short, start)
                      / "Z"             ;Long duration modifier

; une fois au plus, et DigitMapToken et PackagesToken ne sont pas autorisés
; dans la commande AuditCapabilities
auditItem            = ( MuxToken / ModemToken / MediaToken /
                         SignalsToken / EventBufferToken /
                         DigitMapToken / StatsToken / EventsToken /
                         ObservedEventsToken / PackagesToken ) /
                         indAudterminationAudit)

indAudterminationAudit   = indAudauditReturnParameter
                            *(COMMA indAudauditReturnParameter)

indAudauditReturnParameter = (indAudmediaDescriptor / /
                              indAudeventsDescriptor /
                              indAudsignalsDescriptor /
                              indAuddigitMapDescriptor /
                              indAudeventBufferDescriptor /
                              indAudstatisticsDescriptor /
                              indAudpackagesDescriptor)


indAudmediaDescriptor   = MediaToken LBRKT indAudmediaParm RBRKT

; une fois au plus par item
; un parmi streamParm ou streamDescriptor mais pas les deux
```

```
indAudmediaParm            = (indAudstreamParm / indAudstreamDescriptor /
                                 indAudterminationStateDescriptor)

; une fois au plus
indAudstreamParm           = ( indAudlocalControlDescriptor )
; Le traitement SDP est trop complexe pour extraire des éléments individuels
; en vue d'un audit, donc pas d'audit individuel dans le cas local comme dans
; le cas distant

indAudstreamDescriptor     = StreamToken EQUAL StreamID
                               LBRKT indAudstreamParm RBRKT

indAudlocalControlDescriptor = LocalControlToken LBRKT indAudlocalParm RBRKT

; une fois au plus par item
indAudlocalParm            = ( ModeToken / pkgdName /
                               ReservedValueToken /
                               ReservedGroupToken )


indAudterminationStateDescriptor = TerminationStateToken LBRKT
                                 indAudterminationStateParm RBRKT

; une fois au plus par item
indAudterminationStateParm =(pkgdName / ServiceStatesToken / BufferToken)

indAudeventBufferDescriptor = EventBufferToken LBRKT indAudeventSpec RBRKT

indAudeventSpec    = pkgdName [ LBRKT indAudeventSpecParameter RBRKT ]
indAudeventSpecParameter   = (eventStream / eventParameterName)

indAudeventsDescriptor     = EventsToken EQUAL RequestID LBRKT
                              indAudrequestedEvent RBRKT

indAudrequestedEvent       = pkgdName

indAudsignalsDescriptor    = SignalsToken LBRKT [ indAudsignalParm ] RBRKT

indAudsignalParm           = indAudsignalList / indAudsignalRequest

indAudsignalRequest        = signalName
indAudsignalList           = SignalListToken EQUAL signalListId LBRKT
                               indAudsignalListParm RBRKT

indAudsignalListParm    = indAudsignalRequest

indAuddigitMapDescriptor   = DigitMapToken EQUAL (digitMapName )

indAudstatisticsDescriptor = StatsToken LBRKT pkgdName RBRKT

indAudpackagesDescriptor   = PackagesToken LBRKT packagesItem RBRKT


serviceChangeDescriptor = ServicesToken LBRKT serviceChangeParm
                          *(COMMA serviceChangeParm) RBRKT

; chaque paramètre une fois au plus
; au plus un parmi serviceChangeAddress ou serviceChangeMgcId mais
; pas les deux
; serviceChangeMethod et serviceChangeReason sont REQUIS
serviceChangeParm    = (serviceChangeMethod / serviceChangeReason /
                        serviceChangeDelay / serviceChangeAddress /
                        serviceChangeProfile / extension / TimeStamp /
```

```
                      serviceChangeMgcId / serviceChangeVersion /
                      auditItem)

serviceChangeReplyDescriptor = ServicesToken LBRKT
                      servChgReplyParm *(COMMA servChgReplyParm) RBRKT

; une fois au plus. La version est EXIGEE sur la première réponse ServiceChange.
; au plus un parmi serviceChangeAddress ou serviceChangeMgcId mais
; pas les deux
servChgReplyParm     = (serviceChangeAddress / serviceChangeMgcId /
                       serviceChangeProfile / serviceChangeVersion /
                       TimeStamp)
serviceChangeMethod  = MethodToken EQUAL (FailoverToken /
                       ForcedToken / GracefulToken / RestartToken /
                       DisconnectedToken / HandOffToken /
                       extensionParameter)
; serviceChangeReason comporte un code numérique de raison
; et une description alphanumérique facultative.
; serviceChangeReason DOIT être codé au moyen de la forme quotedString de
; value
; quotedString DOIT contenir un code décimal de raison, suivi
; éventuellement d'un blanc unique et d'une chaîne de description
; alphanumérique.
serviceChangeReason  = ReasonToken  EQUAL VALUE
serviceChangeDelay   = DelayToken   EQUAL UINT32
serviceChangeAddress = ServiceChangeAddressToken EQUAL ( mId /
                       portNumber )
serviceChangeMgcId   = MgcIdToken   EQUAL mId
serviceChangeProfile = ProfileToken EQUAL NAME SLASH Version
serviceChangeVersion = VersionToken EQUAL Version
extension            = extensionParameter parmValue

packagesDescriptor   = PackagesToken LBRKT packagesItem
                       *(COMMA packagesItem) RBRKT

Version              = 1*2(DIGIT)
packagesItem         = NAME "-" UINT16

TimeStamp            = Date "T" Time ; per ISO 8601:1988
; Date = yyyymmdd
Date                 = 8(DIGIT)
; Heure = hhmmssss
Time                 = 8(DIGIT)
statisticsDescriptor = StatsToken LBRKT statisticsParameter
                      *(COMMA statisticsParameter ) RBRKT

; une fois au plus par item
statisticsParameter  = pkgdName [EQUAL VALUE]

topologyDescriptor   = TopologyToken LBRKT topologyTriple
                       *(COMMA topologyTriple) RBRKT
topologyTriple       = terminationA COMMA
                       terminationB COMMA topologyDirection
                       [COMMA eventStream ]
terminationA         = TerminationID
terminationB         = TerminationID
topologyDirection    = BothwayToken / IsolateToken / OnewayToken

priority             = PriorityToken EQUAL UINT16

extensionParameter   = "X"  ("-" / "+") 1*6(ALPHA / DIGIT)

; octetString est utilisé pour décrire le protocole SDP dans RFC 2327.
; Il convient d'être vigilant lorsque CRLF dans le RFC 2327 est utilisé.
```

```
; Par sécurité, utiliser EOL dans ce formalisme ABNF.
; Toutes les fois que le caractère "}" apparaît dans le protocole SDP,
; il constitue une séquence d'échappement avec "\", par exemple "\}".
octetString          = *(nonEscapeChar)
nonEscapeChar        = ( "\}" / %x01-7C / %x7E-FF )
; Note – Les guillemets ne sont pas admis dans la forme quotedString.
quotedString         = DQUOTE *(SafeChar / RestChar/ WSP) DQUOTE

UINT16               = 1*5(DIGIT)  ; %x0-FFFF
UINT32               = 1*10(DIGIT) ; %x0-FFFFFFFF

NAME                 = ALPHA *63(ALPHA / DIGIT / "_" )
VALUE                = quotedString / 1*(SafeChar)
SafeChar             = DIGIT / ALPHA / "+" / "-" / "&" /
                       "!" / "_" / "/" / "'" / "?" / "@" /
                       "^" / "`" / "~" / "*" / "$" / "\" /
                       "(" / ")" / "%" / "|" / "."

EQUAL                = LWSP %x3D LWSP ; "="
COLON                = %x3A           ; ":"
LBRKT                = LWSP %x7B LWSP ; "{"
RBRKT                = LWSP %x7D LWSP ; "}"
COMMA                = LWSP %x2C LWSP ; ","
DOT                  = %x2E           ; "."
SLASH                = %x2F           ; "/"
ALPHA                = %x41-5A / %x61-7A ; A-Z / a-z
DIGIT                = %x30-39         ; 0-9
DQUOTE               = %x22            ; " (Double Quote)
HEXDIG               = ( DIGIT / "A" / "B" / "C" / "D" / "E" / "F" )
SP                   = %x20        ; space
HTAB                 = %x09        ; horizontal tab
CR                   = %x0D        ; Carriage return
LF                   = %x0A        ; linefeed
LWSP                 = *( WSP / COMMENT / EOL )
EOL                  = (CR [LF] / LF )
WSP                  = SP / HTAB ; white space
SEP                  = ( WSP / EOL / COMMENT) LWSP
COMMENT              = ";" *(SafeChar/ RestChar / WSP / %x22) EOL
RestChar             = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
                       "<" / ">" / "="

; Les nouveaux jetons ajoutés à sigParameter doivent prendre le format de SPA*
; * peut avoir une forme quelconque, à savoir SPAM
; Les nouveaux jetons ajoutés à eventParameter doivent prendre le format de EPA*
; * peut avoir une forme quelconque, à savoir EPAD

AddToken                  = ("Add"                  / "A")
AuditToken                = ("Audit"                / "AT")
AuditCapToken             = ("AuditCapability"      / "AC")
AuditValueToken           = ("AuditValue"           / "AV")
AuthToken                 = ("Authentication"       / "AU")
BothwayToken              = ("Bothway"              / "BW")
BriefToken                = ("Brief"                / "BR")
BufferToken               = ("Buffer"               / "BF")
CtxToken                  = ("Context"              / "C")
ContextAuditToken         = ("ContextAudit"         / "CA")
DigitMapToken             = ("DigitMap"             / "DM")
DisconnectedToken         = ("Disconnected"         / "DC")
DelayToken                = ("Delay"                / "DL")
DurationToken             = ("Duration"             / "DR")
EmbedToken                = ("Embed"                / "EM")
EmergencyToken            = ("Emergency"            / "EG")
ErrorToken                = ("Error"                / "ER")
EventBufferToken          = ("EventBuffer"          / "EB")
```

```
EventsToken                = ("Events"                / "E")
FailoverToken              = ("Failover"              / "FL")
ForcedToken                = ("Forced"                / "FO")
GracefulToken              = ("Graceful"              / "GR")
H221Token                  = ("H221" )
H223Token                  = ("H223" )
H226Token                  = ("H226" )
HandOffToken               = ("HandOff"               / "HO")
ImmAckRequiredToken        = ("ImmAckRequired"        / "IA")
InactiveToken              = ("Inactive"              / "IN")
IsolateToken               = ("Isolate"               / "IS")
InSvcToken                 = ("InService"             / "IV")
InterruptByEventToken      = ("IntByEvent"            / "IBE")
InterruptByNewSignalsDescrToken
                           = ("IntBySigDescr"         / "IBS")
KeepActiveToken            = ("KeepActive"            / "KA")
LocalToken                 = ("Local"                 / "L")
LocalControlToken          = ("LocalControl"          / "O")
LockStepToken              = ("LockStep"              / "SP")
LoopbackToken              = ("Loopback"              / "LB")
MediaToken                 = ("Media"                 / "M")
MegacopToken               = ("MEGACO"                / "!")
MethodToken                = ("Method"                / "MT")
MgcIdToken                 = ("MgcIdToTry"            / "MG")
ModeToken                  = ("Mode"                  / "MO")
ModifyToken                = ("Modify"                / "MF")
ModemToken                 = ("Modem"                 / "MD")
MoveToken                  = ("Move"                  / "MV")
MTPToken                   = ("MTP")
MuxToken                   = ("Mux"                   / "MX")
NotifyToken                = ("Notify"                / "N")
NotifyCompletionToken      = ("NotifyCompletion"/ "NC")
Nx64kToken                 = ("Nx64Kservice"          / "N64")
ObservedEventsToken        = ("ObservedEvents"        / "OE")
OnewayToken                = ("Oneway"                / "OW")
OnOffToken                 = ("OnOff"                 / "OO")
OtherReasonToken           = ("OtherReason"           / "OR")
OutOfSvcToken              = ("OutOfService"          / "OS")
PackagesToken              = ("Packages"              / "PG")
PendingToken               = ("Pending"               / "PN")
PriorityToken              = ("Priority"              / "PR")
ProfileToken               = ("Profile"               / "PF")
ReasonToken                = ("Reason"                / "RE")
RecvonlyToken              = ("ReceiveOnly"           / "RC")
ReplyToken                 = ("Reply"                 / "P")
RestartToken               = ("Restart"               / "RS")
RemoteToken                = ("Remote"                / "R")
ReservedGroupToken         = ("ReservedGroup"         / "RG")
ReservedValueToken         = ("ReservedValue"         / "RV")
SendonlyToken              = ("SendOnly"              / "SO")
SendrecvToken              = ("SendReceive"           / "SR")
ServicesToken              = ("Services"              / "SV")
ServiceStatesToken         = ("ServiceStates"         / "SI")
ServiceChangeToken         = ("ServiceChange"         / "SC")
ServiceChangeAddressToken  = ("ServiceChangeAddress"  / "AD")
SignalListToken            = ("SignalList"            / "SL")
SignalsToken               = ("Signals"               / "SG")
SignalTypeToken            = ("SignalType"            / "SY")
StatsToken                 = ("Statistics"            / "SA")
StreamToken                = ("Stream"                / "ST")
SubtractToken              = ("Subtract"              / "S")
SynchISDNToken             = ("SynchISDN"             / "SN")
TerminationStateToken      = ("TerminationState"      / "TS")
TestToken                  = ("Test"                  / "TE")
```

```
TimeOutToken              = ("TimeOut"               / "TO")
TopologyToken             = ("Topology"              / "TP")
TransToken                = ("Transaction"           / "T")
ResponseAckToken          = ("TransactionResponseAck"/ "K")
V18Token                  = ("V18")
V22Token                  = ("V22")
V22bisToken               = ("V22b")
V32Token                  = ("V32")
V32bisToken               = ("V32b")
V34Token                  = ("V34")
V76Token                  = ("V76")
V90Token                  = ("V90")
V91Token                  = ("V91")
VersionToken              = ("Version"               /   "V")
```

B.3 Codage hexadécimal des octets

Le codage hexadécimal des octets est un moyen de représenter une chaîne d'octets comme une chaîne de chiffres hexadécimaux, chaque octet étant représenté par deux chiffres. Ce codage des octets devrait être employé lors du codage de chaînes d'octets dans la version alphanumérique du protocole.

Pour chaque octet, la séquence à 8 bits est codée au moyen de chiffres hexadécimaux. Le bit 0 est transmis en premier lieu, tandis que le bit 7 est transmis en dernier lieu.

Les bits 7 à 4 sont codés comme étant le premier chiffre hexadécimal, le bit 7 étant le bit de poids le plus fort et le bit 4 étant celui de poids le plus faible. Les bits 3 à 0 sont codés comme étant le second chiffre hexadécimal, le bit 3 étant le bit de poids le plus fort et le bit 0 étant celui de poids le plus faible.

Exemples:

Forme binaire des octets	Codage hexadecimal
00011011	D8
11100100	27
10000011 10100010 11001000 00001001	C1451390

B.4 Séquence hexadécimale pour les octets

Une séquence hexadécimale pour les octets est un nombre pair de chiffres hexadécimaux, s'achevant par un caractère <CR>.

Annexe C

Etiquettes des propriétés de flux média

Les paramètres des descripteurs Local, Remote et LocalControl sont spécifiés sous forme de paires de valeurs d'étiquette si le codage binaire est utilisé pour le protocole. La présente annexe contient les noms de propriétés (PropertyID), les étiquettes de propriétés (Property tag), le type de la propriété (Type) et les valeurs (Value). Les valeurs fournies dans le champ Value lorsque le champ contient des références doivent être considérées comme une "information". La référence contient les valeurs normatives. Si un champ de valeur ne contient pas de référence, les valeurs de ce champ doivent être considérées comme "normatives".

Dans la présente annexe, les étiquettes sont exprimées en nombres hexadécimaux. Lors du réglage de la valeur d'une propriété, un contrôleur MGC peut sous-spécifier la valeur selon l'un des mécanismes spécifiés au § 7.1.1.

La prise en charge des propriétés de la présente annexe ou de ses paragraphes est facultative. Par exemple, seules trois propriétés de C.3 et cinq propriétés de C.8 pourraient être appliquées.

Pour le type "énumération", la valeur est représentée par la valeur entre parenthèses, par exemple Send(0), Receive(1). Les propriétés à l'Annexe C pour les types à "N bits" ou à "M octets" devraient être traitées comme des chaînes d'octets lors du codage du protocole, tandis que les propriétés à "N bits entier" seront traitées comme des entiers. "String" sera traitée comme une chaîne IA5String lors du codage du protocole.

Lorsqu'un type présente une valeur inférieure à un octet, cette valeur doit être enregistrée dans les bits de plus faible poids d'une chaîne d'octet de taille 1.

C.1 Attributs généraux des médias

Identificateur de propriété	Etiquette de propriété	Type	Valeur
Média	1001	Enumération	Audio(0), Video(1), Data(2)
Mode de transmission	1002	Enumération	Send(0), Receive(1), Send&Receive(2)
Nombre de voies	1003	Entier non signé	0-255
Fréquence d'échantillonnage	1004	Entier non signé	0-2^32
Débit	1005	Integer (entier)	(0..4294967295) NOTE – Unités de 100 bit/s.
Codec A	1006	Octet String (chaîne d'octets)	Audio Type de codec: Réf.: Rec. UIT-T Q.765.5 – Mécanisme de transport d'application. Les codecs non-UIT-T sont définis avec une organisation de normes appropriée sous un identificateur organisationnel défini.
Samplepp	1007	Entier non signé	Nombre maximal d'échantillons ou de trames pour chaque paquet: 0..65535
Silencesupp	1008	Booléen	Suppression de silence: Vrai/Faux
Encrypttype	1009	Octet String (chaîne d'octets)	Réf.: Rec. UIT-T H.245
Encryptkey	100A	Octet string size (0..65535)	Encryption key (clé de cryptage) Réf.: Rec. UIT-T H.235
Echocanc	100B		Non utilisée. Voir § E.13 pour un exemple de possibles propriétés de contrôle d'écho.
Gain	100C	Entier non signé	Gain en dB: 0..65535
Jitterbuff	100D	Entier non signé	Taille du tampon de gigue ms: 0..65535
PropDelay	100E	Entier non signé	Retard de propagation: 0..65535 Temps de propagation maximal en millisecondes pour la connexion support entre deux passerelles MG. Le temps maximal dépendra de la technique de support.
RTPpayload	100F	Entier	Type de charge utile dans le Profil RTP pour conférences audio et vidéo avec commandes minimales Réf.: RFC 1890

C.2 Propriétés Mux

Identificateur de propriété	Etiquette de propriété	Type	Valeur
H222	2001	Octet String (chaîne d'octets)	H222LogicalChannelParameters Réf.: Rec. UIT-T H.245
H223	2002	Octet String (chaîne d'octets)	H223LogicalChannelParameters Réf.: Rec. UIT-T H.245
V76	2003	Octet String (chaîne d'octets)	V76LogicalChannelParameters Réf.: Rec. UIT-T H.245
H2250	2004	Octet String (chaîne d'octets)	H2250LogicalChannelParameters Réf.: Rec. UIT-T H.245

C.3 Propriétés générales de support

Identificateur de propriété	Etiquette de propriété	Type	Valeur
Mediatx	3001	Enumération	Type de transport média Circuit(0) TDM, ATM(1), FR(2), Ipv4(3), Ipv6(4), …
BIR	3002	4 octets	La valeur dépend de la technique de transport
NSAP	3003	1-20 octets	Voir NSAP. Réf.: Annexe A/X.213

C.4 Propriétés générales du mode ATM

Identificateur de propriété	Etiquette de propriété	Type	Valeur
AESA	4001	20 octets	Adresse de système d'extrémité ATM
VPVC	4002	4 octets: VPCI dans les deux premiers octets de poids le plus faible, VCI dans les deux octets suivants	VPCI/VCI Réf.: Rec. UIT-T Q.2931
SC	4003	Enumération	Catégorie de service CBR(0), nrt-VBR1(1), nrt-VBR2(2), nrt-VBR3(3), rt-VBR1(4), rt-VBR2(5), rt-VBR3(6), UBR1(7), UBR2(8), ABR(9). Réf.: Forum ATM UNI 4.0
BCOB	4004	Entier de 5 bits	Classe de support large bande Réf.: Rec. UIT-T Q.2961.2
BBTC	4005	Entier de 7 bits	Capacité de transfert large bande Réf.: Rec. UIT-T Q.2961.1

Identificateur de propriété	Etiquette de propriété	Type	Valeur
ATC	4006	Enumération	Capacité de trafic ATM I.371 DBR(0), SBR1(1), SBR2(2), SBR3(3), ABT/IT(4), ABT/DT(5), ABR(6) Réf.: Rec. UIT-T I.371
STC	4007	2 bits	Sensibilité à la mutilation: Bits 2 1 0 0 insensible à la mutilation 0 1 sensible à la mutilation Réf.: Rec. UIT-T Q.2931
UPCC	4008	2 bits	Configuration de la connexion de plan utilisateur (*user plane connection*): Bits 2 1 0 0 point à point 0 1 point à multipoint Réf.: Rec. UIT-T Q.2931
PCR0	4009	Entier de 24 bits	Débit cellulaire crête (pour CLP = 0) Réf.: Rec. UIT-T Q.2931
SCR0	400A	Entier de 24 bits	Débit soutenable (pour CLP = 0) Réf.: Rec. UIT-T Q.2961.1
MBS0	400B	Entier de 24 bits	Taille maximale des rafales (pour CLP = 0) Réf.: Rec. UIT-T Q.2961.1
PCR1	400C	Entier de 24 bits	Débit cellulaire crête (pour CLP = 0 + 1) Réf.: Rec. UIT-T Q.2931
SCR1	400D	Entier de 24 bits	Débit soutenable (pour CLP = 0 + 1) Réf.: Rec. UIT-T Q.2961.1
MBS1	400E	Entier de 24 bits	Taille maximale des rafales (pour CLP = 0 + 1) Réf.: Rec. UIT-T Q.2961.1
BEI	400F	Booléen	Indicateur de connexion au mieux La valeur 1 indique que BEI doit être inclus dans la signalisation ATM; la valeur 0 indique que BEI ne doit pas être inclus dans la signalisation ATM. Réf.: Forum ATM UNI 4.0
TI	4010	Booléen	Indicateur d'étiquetage La valeur 0 indique que l'étiquetage n'est pas autorisé; La valeur 1 indique que l'étiquetage est demandé. Réf.: Rec. UIT-T Q.2961.1

<table>
<tr><th>Identificateur de propriété</th><th>Etiquette de propriété</th><th>Type</th><th>Valeur</th></tr>
<tr><td>FD</td><td>4011</td><td>Booléen</td><td>Mise à l'écart de trame
La valeur 0 indique qu'aucune mise à l'écart de trame n'est autorisée;
la valeur 1 indique que la mise à l'écart de trame est autorisée.
Réf.: Forum ATM UNI 4.0</td></tr>
<tr><td>A2PCDV</td><td>4012</td><td>Entier de 24 bits</td><td>CDV 2 points acceptable
Réf.: Rec. UIT-T Q.2965.2</td></tr>
<tr><td>C2PCDV</td><td>4013</td><td>Entier de 24 bits</td><td>CDV 2 points cumulé
Réf.: Rec. UIT-T Q.2965.2</td></tr>
<tr><td>APPCDV</td><td>4014</td><td>Entier de 24 bits</td><td>CDV P-P acceptable
Réf.: Forum ATM UNI 4.0</td></tr>
<tr><td>CPPCDV</td><td>4015</td><td>Entier de 24 bits</td><td>CDV P-P cumulé
Réf.: Forum ATM UNI 4.0</td></tr>
<tr><td>ACLR</td><td>4016</td><td>Entier de 8 bits</td><td>Taux de perte de cellule acceptable
Réf.: Rec. UIT-T Q.2965.2, Forum ATM UNI 4.0</td></tr>
<tr><td>MEETD</td><td>4017</td><td>Entier de 16 bits</td><td>Temps de transit de bout en bout maximal
Réf.: Rec. UIT-T Q.2965.2, Forum ATM UNI 4.0</td></tr>
<tr><td>CEETD</td><td>4018</td><td>Entier de 16 bits</td><td>Temps de transit de bout en bout cumulé
Réf.: Rec. UIT-T Q.2965.2, Forum ATM UNI 4.0</td></tr>
<tr><td>QosClass</td><td>4019</td><td>Entier compris entre 0 et 5</td><td>Classe de QS
<table><tr><th>Classe de QS</th><th>Signification</th></tr><tr><td>0</td><td>QS par défaut associée à la capacité de transfert ATC telle que définie dans la Rec. UIT-T Q.2961.2</td></tr><tr><td>1</td><td>Strict</td></tr><tr><td>2</td><td>Tolérant</td></tr><tr><td>3</td><td>A deux niveaux</td></tr><tr><td>4</td><td>Non borné</td></tr><tr><td>5</td><td>Strict à deux niveaux</td></tr></table>Réf.: Rec. UIT-T Q.2965.1</td></tr>
<tr><td>AALtype</td><td>401A</td><td>1 octet</td><td>Type de couche AAL
Bits
<u>8 7 6 5 4 3 2 1</u>
0 0 0 0 0 0 0 0 AAL pour voix
0 0 0 0 0 0 0 1 AAL type 1
0 0 0 0 0 0 1 0 AAL type 2
0 0 0 0 0 0 1 1 AAL type 3/4
0 0 0 0 0 1 0 1 AAL type 5
0 0 0 1 0 0 0 0 AAL définie par utilisateur
Réf.: Rec. UIT-T Q.2931</td></tr>
</table>

C.5 Relais de trames

Identificateur de propriété	Etiquette de propriété	Type	Valeur
DLCI	5001	Entier non signé	Identificateur de connexion de liaison de données
CID	5002	Entier non signé	sub-channel id (identificateur de sous-voie)
SID/Noiselevel	5003	Entier non signé	Descripteur d'insertion de silence
Primary Payload Type	5004	Entier non signé	Type de charge utile primaire couvre FAX et codecs

C.6 IP

Identificateur de propriété	Etiquette de propriété	Type	Valeur
IPv4	6001	32 bits Ipv4Address	Adresse IPv4 Réf.: IETF RFC 791
IPv6	6002	128 bits	Adresse IPv6 Réf.: IETF RFC 2460
Port	6003	Entier non signé	0..65535
Porttype	6004	Enuméré	TCP(0), UDP(1), SCTP(2)

C.7 ATM AAL 2

Identificateur de propriété	Etiquette de propriété	Type	Valeur
AESA	7001	20 octets	Adresse de point d'extrémité de service AAL 2 comme défini dans la Recommandation référencée. ESEA NSEA Réf.: Rec. UIT-T Q.2630.1
BIR	Voir § C.3	4 octets	Réf. produite par utilisateur du service comme défini dans la Recommandation référencée. SUGR Réf.: Rec. UIT-T Q.2630.1
ALC	7002	12 octets	Caractéristiques de liaison AAL 2 comme défini dans la Recommandation référencée. Débit CPS-SDU max./moyen Longueur d'unité CPS-SDU max./moyenne Réf.: Rec. UIT-T Q.2630.1

Identificateur de propriété	Etiquette de propriété	Type	Valeur
SSCS	7003	I.366.2: audio (8 octets); multidébit (3 octets), ou I.366.1: SAR-assuré (14 octets); SAR non assuré (7 octets).	Informations de sous-couche de convergence propres au service comme défini dans: – UIT-T Q.2630.1, et utilisé dans: – UIT-T I.366.2: audio/multidébit; – UIT-T I.366.1: SAR-assuré/non assuré. Réf.: Recs. UIT-T Q.2630.1, I.366.1 et I.366.2
SUT	7004	1..254 octets	Paramètre de transport d'utilisateur du service comme défini dans la Recommandation référencée. Réf.: Rec. UIT-T Q.2630.1
TCI	7005	Booléen	Indicateur de connexion d'essai comme défini dans la Recommandation référencée. Réf.: Rec. UIT-T Q.2630.1
Timer_CU	7006	Entier de 32 bits	Temporisateur-CU Millisecondes de maintien de cellule partiellement remplie avant émission.
MaxCPSSDU	7007	Entier de 8 bits	Unité de données de service CPS maximale Réf.: Rec. UIT-T Q.2630.1
CID	7008	8 bits	Subchannel id (id de sous-voie): 0-255 Réf.: Rec. UIT-T I.363.2

C.8 ATM AAL 1

Identificateur de propriété	Etiquette de propriété	Type	Valeur
BIR	Voir tableau au § C.3	4-29 octets	Transport d'identificateur générique (GIT, *generic identifier transport*) Réf.: Rec. UIT-T Q.2941.1
AAL1ST	8001	1 octet	Sous-type AAL 1 Bits 8 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 néant 0 0 0 0 0 0 0 1 transport de signaux en bande vocale à 64 kbit/s 0 0 0 0 0 0 1 0 transport en mode circuit 0 0 0 0 0 1 0 0 transport de signaux audio de haute qualité 0 0 0 0 0 1 0 1 transport de signaux vidéo Réf.: Rec. UIT-T Q.2931

Identificateur de propriété	Etiquette de propriété	Type	Valeur
CBRR	8002	1 octet	Débit CBR Bits 8 7 6 5 4 3 2 1 0 0 0 0 0 0 0 1 64 kbit/s 0 0 0 0 0 1 0 0 1544 kbit/s 0 0 0 0 0 1 0 1 6312 kbit/s 0 0 0 0 0 1 1 0 32 064 kbit/s 0 0 0 0 0 1 1 1 44 736 kbit/s 0 0 0 0 1 0 0 0 97 728 kbit/s 0 0 0 1 0 0 0 0 2048 kbit/s 0 0 0 1 0 0 0 1 8448 kbit/s 0 0 0 1 0 0 1 0 34 368 kbit/s 0 0 0 1 0 0 1 1 139 264 kbit/s 0 1 0 0 0 0 0 0 n × 64 kbit/s 0 1 0 0 0 0 0 1 n × 8 kbit/s Réf.: Rec. UIT-T Q.2931
MULT	Voir tableau au § C.9		Multiplicateur de circuits, ou n × 64k/8k/300 Réf.: Rec. UIT-T Q.2931
SCRI	8003	1 octet	Méthode de recalage sur la fréquence d'horloge source Bits 8 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 néant 0 0 0 0 0 0 0 1 SRTS 0 0 0 0 0 0 1 0 ACM Réf.: Rec. UIT-T Q.2931
ECM	8004	1 octet	Méthode de correction d'erreur Bits 8 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 néant 0 0 0 0 0 0 0 1 FEC – Pertes 0 0 0 0 0 0 1 0 FEC – Temps Réf.: Rec. UIT-T Q.2931
SDTB	8005	Entier de 16 bits	Longueur de bloc de transfert de données structurées Longueur de bloc de service SDT CBR Réf.: Rec. UIT-T I.363.1
PFCI	8006	Entier de 8 bits	Identificateur de cellules partiellement remplies 1-47 Réf.: Rec. UIT-T I.363.1

C.9 Capacités du circuit support

Les entrées du tableau qui font référence à la Rec. UIT-T Q.931 concerne le codage de l'élément d'information capacité support de la Rec. UIT-T Q.931 et non pas l'élément d'information de couche inférieure.

Identificateur de propriété	Etiquette de propriété	Type	Valeur
TMR	9001	1 octet	Support de transmission requis (Rec. UIT-T Q.763) Bits 8 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 parole 0 0 0 0 0 0 0 1 libre 0 0 0 0 0 0 1 0 64 kbit/s sans restriction 0 0 0 0 0 0 1 1 3,1 kHz audio 0 0 0 0 0 1 0 0 réservé pour une voix de remplacement (service 2)/ 64 kbit/s sans restriction (service 1) 0 0 0 0 0 1 0 1 réservé pour d'autres 64 kbit/s sans restriction (service 1)/ parole (service 2) 0 0 0 0 0 1 1 0 64 kbit/s de préférence 0 0 0 0 0 1 1 1 2 × 64 kbit/s sans restriction 0 0 0 0 1 0 0 0 384 kbit/s sans restriction 0 0 0 0 1 0 0 1 1536 kbit/s sans restriction 0 0 0 0 1 0 1 0 1920 kbit/s sans restriction 0 0 0 0 1 0 1 1 à réservé 0 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0 3 × 64 kbit/s sans restriction 0 0 0 1 0 0 0 1 4 × 64 kbit/s sans restriction 0 0 0 1 0 0 1 0 5 × 64 kbit/s sans restriction 0 0 0 1 0 0 1 1 réservé 0 0 0 1 0 1 0 0 7 × 64 kbit/s sans restriction 0 0 0 1 0 1 0 1 8 × 64 kbit/s sans restriction 0 0 0 1 0 1 1 0 9 × 64 kbit/s sans restriction 0 0 0 1 0 1 1 1 10 × 64 kbit/s sans restriction 0 0 0 1 1 0 0 0 11 × 64 kbit/s sans restriction 0 0 0 1 1 0 0 1 12 × 64 kbit/s sans restriction 0 0 0 1 1 0 1 0 13 × 64 kbit/s sans restriction 0 0 0 1 1 0 1 1 14 × 64 kbit/s sans restriction 0 0 0 1 1 1 0 0 15 × 64 kbit/s sans restriction 0 0 0 1 1 1 0 1 16 × 64 kbit/s sans restriction 0 0 0 1 1 1 1 0 17 × 64 kbit/s sans restriction 0 0 0 1 1 1 1 1 18 × 64 kbit/s sans restriction 0 0 1 0 0 0 0 0 19 × 64 kbit/s sans restriction 0 0 1 0 0 0 0 1 20 × 64 kbit/s sans restriction 0 0 1 0 0 0 1 0 21 × 64 kbit/s sans restriction 0 0 1 0 0 0 1 1 22 × 64 kbit/s sans restriction 0 0 1 0 0 1 0 0 23 × 64 kbit/s sans restriction 0 0 1 0 0 1 0 1 réservé 0 0 1 0 0 1 1 0 25 × 64 kbit/s sans restriction 0 0 1 0 0 1 1 1 26 × 64 kbit/s sans restriction 0 0 1 0 1 0 0 0 27 × 64 kbit/s sans restriction 0 0 1 0 1 0 0 1 28 × 64 kbit/s sans restriction 0 0 1 0 1 0 1 0 29 × 64 kbit/s sans restriction 0 0 1 0 1 0 1 1 à réservé 1 1 1 1 1 1 1 1 Réf.: Rec. UIT-T Q.763

Identificateur de propriété	Etiquette de propriété	Type	Valeur
TMRSR	9002	1 octet	Débit sous-multiple du support de transmission requis 0 non spécifié 1 8 kbit/s 2 16 kbit/s 3 32 kbit/s
Contcheck	9003	Booléen	Contrôle de continuité 0 non requis sur ce circuit 0 requis sur ce circuit Réf.: Rec. UIT-T Q.763
ITC	9004	5 bits	Capacité de transfert d'informations Bits 5 4 3 2 1 0 0 0 0 0 parole 0 1 0 0 0 informations numériques sans restriction 0 1 0 0 1 informations numériques avec restriction 1 0 0 0 0 3,1 kHz audio 1 0 0 0 1 informations numériques sans restriction avec tonalités/annonces 1 1 0 0 0 vidéo Toutes les autres valeurs sont réservées. Réf.: Rec. UIT-T Q.763
TransMode	9005	2 bits	Mode de transfert Bits 2 1 0 0 mode Circuit 1 0 mode Packet Réf.: Rec. UIT-T Q.931
TransRate	9006	5 bits	Débit de transfert Bits 5 4 3 2 1 0 0 0 0 0 ce code doit être utilisé pour les appels en mode paquet 1 0 0 0 0 64 kbit/s 1 0 0 0 1 2×64 kbit/s 1 0 0 1 1 384 kbit/s 1 0 1 0 1 1536 kbit/s 1 0 1 1 1 1920 kbit/s 1 1 0 0 0 multidébit (64 kbit/s débit de base) Réf.: Rec. UIT-T Q.931
MULT	9007	7 bits	Multiplicateur de débit Toute valeur de 2 à n (nombre maximal de voie B) Réf.: Rec. UIT-T Q.931

Identificateur de propriété	Etiquette de propriété	Type	Valeur
layer1prot	9008	5 bits	Protocole de couche 1 des informations d'utilisateur Bits 5 4 3 2 1 0 0 0 0 1 adaptation de débit normalisé UIT-T V.110 et X.30. 0 0 0 1 0 Rec. UIT-T G.711 Loi μ 0 0 0 1 1 Rec. UIT-T G.711 Loi A 0 0 1 0 0 Rec. UIT-T G.726 32 kbit/s MICDA et Rec. UIT-T I.460 0 0 1 0 1 Recs. UIT-T H.221 et H.242 0 0 1 1 0 Recs. UIT-T H.223 et H.245 0 0 1 1 1 adaptation de débit normalisé non-UIT-T. 0 1 0 0 0 adaptation de débit normalisé UIT-T V.120. 0 1 0 0 1 adaptation de débit normalisé UIT-T X.31 avec remplissage par fanions HDLC Toutes les autres valeurs sont réservées. Réf.: Rec. UIT-T Q.931
syncasync	9009	Booléen	Synchrone/Asynchrone 0 données synchrones 1 données asynchrones Réf.: Rec. UIT-T Q.931
negotiation	900A	Booléen	Négociation 0 négociation dans la bande possible 1 négociation dans la bande impossible Réf.: Rec. UIT-T Q.931

Identificateur de propriété	Etiquette de propriété	Type	Valeur
Userrate	900B	5 bits	Débit d'utilisateur Bits 5 4 3 2 1 0 0 0 0 0 le débit est indiqué en bits E spécifiés dans la Rec. UIT-T I.460 ou il peut être négocié dans la bande 0 0 0 0 1 0,6 kbit/s Recs. UIT-T V.6 et X.1 0 0 0 1 0 1,2 kbit/s Rec. UIT-T V.6 0 0 0 1 1 2,4 kbit/s Recs. UIT-T V.6 et X.1 0 0 1 0 0 3,6 kbit/s Rec. UIT-T V.6 0 0 1 0 1 4,8 kbit/s Recs. UIT-T V.6 et X.1 0 0 1 1 0 7,2 kbit/s Rec. UIT-T V.6 0 0 1 1 1 8 kbit/s Rec. UIT-T I.460 0 1 0 0 0 9,6 kbit/s Recs. UIT-T V.6 et X.1 0 1 0 0 1 14,4 kbit/s Rec. UIT-T V.6 0 1 0 1 0 16 kbit/s Rec. UIT-T I.460 0 1 0 1 1 19,2 kbit/s Rec. UIT-T V.6 0 1 1 0 0 32 kbit/s Rec. UIT-T I.460 0 1 1 0 1 38,4 kbit/s Rec. UIT-T V.110 0 1 1 1 0 48 kbit/s Recs. UIT-T V.6 et X.1 0 1 1 1 1 56 kbit/s Rec. UIT-T V.6 1 0 0 1 0 57,6 kbit/s Rec. UIT-T V.14 étendue 1 0 0 1 1 28,8 kbit/s Rec. UIT-T V.110 1 0 1 0 0 24 kbit/s Rec. UIT-T V.110 1 0 1 0 1 0,1345 kbit/s Rec. UIT-T X.1 1 0 1 1 0 0,100 kbit/s Rec. UIT-T X.1 1 0 1 1 1 0,075/1,2 kbit/s Recs. UIT-T V.6 et X.1 1 1 0 0 0 1,2/0,075 kbit/s Recs. UIT-T V.6 et X.1 1 1 0 0 1 0,050 kbit/s Recs. UIT-T V.6 et X.1 1 1 0 1 0 0,075 kbit/s Recs. UIT-T V.6 et X.1 1 1 0 1 1 0,110 kbit/s Recs. UIT-T V.6 et X.1 1 1 1 0 0 0,150 kbit/s Recs. UIT-T V.6 et X.1 1 1 1 0 1 0,200 kbit/s Recs. UIT-T V.6 et X.1 1 1 1 1 0 0,300 kbit/s Recs. UIT-T V.6 et X.1 1 1 1 1 1 12 kbit/s Rec. UIT-T V.6 Toutes les autres valeurs sont réservées. Réf.: Rec. UIT-T Q.931
INTRATE	900C	2 bits	Débit intermédiaire Bits 2 1 0 0 inutilisé 0 1 8 kbit/s 1 0 16 kbit/s 1 1 32 kbit/s Réf.: Rec. UIT-T Q.931

Identificateur de propriété	Etiquette de propriété	Type	Valeur
nictx	900D	Booléen	Horloge indépendante du réseau (NIC, *network independent clock*) à l'émission 0 non requis pour envoyer des données avec horloge indépendante du réseau 1 requis pour envoyer données avec horloge indépendante du réseau Réf.: Rec. UIT-T Q.931
nicrx	900E	Booléen	Horloge indépendante du réseau (NIC) à la réception 0 ne peut pas accepter de données avec horloge indépendante du réseau (c'est-à-dire que l'émetteur ne prend pas en charge cette procédure facultative) 1 peut accepter des données avec horloge indépendante du réseau (c'est-à-dire que l'émetteur prend bien en charge cette procédure facultative) Réf.: Rec. UIT-T Q.931
flowconttx	900F	Booléen	Commande de débit à l'émission (Tx, *transmission*) 0 non requis pour envoyer données avec mécanisme de commande de débit 1 requis pour envoyer données avec mécanisme de commande de débit Réf.: Rec. UIT-T Q.931
flowcontrx	9010	Booléen	Commande de débit à la réception (Rx, *reception*) 0 ne peut pas accepter de données avec mécanisme de commande de débit (c'est-à dire que l'émetteur ne prend pas en charge cette procédure facultative) 1 peut accepter des données avec mécanisme de commande de débit (c'est-à-dire que l'émetteur prend bien en charge cette procédure facultative) Réf.: Rec. UIT-T Q.931
rateadapthdr	9011	Booléen	En-tête/pas d'en-tête d'adaptation de débit 0 en-tête d'adaptation de débit non inclus 1 en-tête d'adaptation de débit inclus Réf.: Rec. UIT-T Q.931
multiframe	9012	Booléen	Prise en charge de l'établissement de multitrames dans la liaison de données 0 établissement de multitrames non pris en charge. Seules les trames UI sont autorisées 1 prise en charge de l'établissement de multitrames Réf.: Rec. UIT-T Q.931

Identificateur de propriété	Etiquette de propriété	Type	Valeur
OPMODE	9013	Booléen	Mode de fonctionnement 0 mode de fonctionnement transparent aux bits 1 mode de fonctionnement sensible au protocole Réf.: Rec. UIT-T Q.931
llidnegot	9014	Booléen	Négociation d'identificateur de liaison logique 0 valeur par défaut, LLI = 256 seulement 1 négociation de protocole complète Réf.: Rec. UIT-T Q.931
assign	9015	Booléen	Commettant/Commissionnaire 0 l'expéditeur du message est "attribué par défaut" 1 l'expéditeur du message est "attributeur seulement" Réf.: Rec. UIT-T Q.931
inbandneg	9016	Booléen	Négociation dans la bande/hors bande 0 négociation effectuée avec les messages INFORMATION UTILISATEUR sur une connexion de signalisation temporaire 1 la négociation est effectuée dans la bande au moyen de la liaison logique zéro Réf.: Rec. UIT-T Q.931
stopbits	9017	2 bits	Nombre de bits d'arrêt Bits 2 1 0 0 inutilisé 0 1 1 bit 1 0 1,5 bit 1 1 2 bits Réf.: Rec. UIT-T Q.931
databits	9018	2 bits	Nombre de données bits, non compris le bit de parité si présent Bits 2 1 0 0 inutilisé 0 1 5 bits 1 0 7 bits 1 1 8 bits Réf.: Rec. UIT-T Q.931

Identificateur de propriété	Etiquette de propriété	Type	Valeur
parity	9019	3 bits	Informations de parité Bits 3 2 1 0 0 0 impaire 0 1 0 paire 0 1 1 aucune 1 0 0 forcée à 0 1 0 1 forcée à 1 Toutes les autres valeurs sont réservées. Réf.: Rec. UIT-T Q.931
duplexmode	901A	Booléen	Mode duplex 0 semi-duplex 1 duplex Réf.: Rec. UIT-T Q.931
modem	901B	6 bits	Type de modem Bits 6 5 4 3 2 1 0 0 0 0 0 0 à usage national 0 0 0 1 0 1 0 1 0 0 0 1 Rec. UIT-T V.21 0 1 0 0 1 0 Rec. UIT-T V.22 0 1 0 0 1 1 Rec. UIT-T V.22 *bis* 0 1 0 1 0 0 Rec. UIT-T V.23 0 1 0 1 0 1 Rec. UIT-T V.26 0 1 1 0 0 1 Rec. UIT-T V.26 *bis* 0 1 0 1 1 1 Rec. UIT-T V.26 *ter* 0 1 1 0 0 0 Rec. UIT-T V.27 0 1 1 0 0 1 Rec. UIT-T V.27 *bis* 0 1 1 0 1 0 Rec. UIT-T V.27 *ter* 0 1 1 0 1 1 Rec. UIT-T V.29 0 1 1 1 0 1 Rec. UIT-T V.32 0 1 1 1 1 0 Rec. UIT-T V.34 1 0 0 0 0 0 à usage national 1 0 1 1 1 1 1 1 0 0 0 0 à spécifié par l'utilisateur 1 1 1 1 1 1 Réf.: Rec. UIT-T Q.931

Identificateur de propriété	Etiquette de propriété	Type	Valeur
layer2prot	901C	5 bits	Protocole de couche 2 des informations d'utilisateur Bits 5 4 3 2 1 0 0 0 1 0 Rec. UIT-T Q.921/I.441 0 0 1 1 0 Rec. UIT-T X.25, couche Liaison 0 1 1 0 0 commande de liaison logique de réseau local (ISO/CEI 8802-2) Toutes les autres valeurs sont réservées. Réf.: Rec. UIT-T Q.931
layer3prot	901D	5 bits	Protocole de couche 3 des informations d'utilisateur Bits 5 4 3 2 1 0 0 0 1 0 Rec. UIT-T Q.931 0 0 1 1 0 Rec. UIT-T X.25, couche paquet 0 1 0 1 1 ISO/CEI TR 9577 (identification de protocole dans la couche Réseau) Toutes les autres valeurs sont réservées. Réf.: Rec. UIT-T Q.931
addlayer3prot	901E	Octet	Protocole de couche 3 additionnel des informations d'utilisateur Bits 4 3 2 1 4 3 2 1 1 1 0 0 1 1 0 0 protocole Internet (RFC 791) (ISO/CEI TR 9577) 1 1 0 0 1 1 1 1 protocole point à point (RFC 1661) Réf.: Rec. UIT-T Q.931
DialledN	901F	30 octets	Numéro appelé
DiallingN	9020	30 octets	Numéro appelant
ECHOCI	9021		Non employé. Voir un exemple des propriétés possibles de limitation d'écho au § E.13

Identificateur de propriété	Etiquette de propriété	Type	Valeur
NCI	9022	1 octet	Nature des indicateurs de connexion Bits 2 1 *indicateur de satellite* 0 0 aucun circuit satellite dans la connexion 0 1 un seul circuit satellite dans la connexion 1 0 deux circuits satellites dans la connexion 1 1 réservé Bits 4 3 *indicateur de contrôle de continuité* 0 0 contrôle de continuité non requis 0 1 contrôle de continuité requis sur ce circuit 1 0 contrôle de continuité effectué sur un circuit précédent 1 1 réservé Bits 5 *indicateur de limiteur d'écho* 0 limiteur d'écho sortant non inclus 1 limiteur d'écho sortant inclus Bits 8 7 6 champ réservé Réf.: Rec. UIT-T Q.763
USI	9023	Octet string	Information relative au service utilisateur Réf.: § 3.57/Q.763

C.10 Propriété AAL 5

Identificateur de propriété	Etiquette de propriété	Type	Valeur
FMSDU	A001	Entier de 32 bits	Longueur maximale d'unité CPCS-SDU vers l'avant: Longueur maximale d'unité CPCS-SDU envoyée dans le sens de l'appelant vers l'appelé. Réf.: Rec. UIT-T Q.2931
BMSDU	A002	Entier de 32 bits	Longueur maximale d'unité CPCS-SDU vers l'arrière: Longueur maximale d'unité CPCS-SDU envoyée dans le sens de l'appelé vers l'appelant. Réf.: Rec. UIT-T Q.2931
SSCS	voir tableau au § C.7	voir tableau au § C.7	Voir tableau au C.7 pour des valeurs supplémentaires: VPI/VCI

C.11 Equivalents SDP

Identificateur de propriété	Etiquette de propriété	Type	Valeur
SDP_V	B001	String	Version du protocole Réf.: RFC 2327
SDP_O	B002	String	ID de propriétaire/créateur et session Réf.: RFC 2327
SDP_S	B003	String	Nom de session Réf.: RFC 2327
SDP_I	B004	String	Identificateur de session Réf.: RFC 2327
SDP_U	B005	String	URI de descripteur Réf.: RFC 2327
SDC_E	B006	String	Adresse email Réf.: RFC 2327
SDP_P	B007	String	Numéro de téléphone Réf.: RFC 2327
SDP_C	B008	String	Informations de connexion Réf.: RFC 2327
SDP_B	B009	String	Informations de largeur de bande Réf.: RFC 2327
SDP_Z	B00A	String	Réglage de la zone horaire Réf.: RFC 2327
SDP_K	B00B	String	Clé de cryptage Réf.: RFC 2327
SDP_A	B00C	String	Zéro ou plus d'attributs de session Réf.: RFC 2327
SDP_T	B00D	String	Durée de session active Réf.: RFC 2327
SDP_R	B00E	String	Zéro ou plus de répétitions Réf.: RFC 2327
SDP_M	B00F	String	Type de média, port, transport et format Réf.: RFC 2327

C.12 H.245

Identificateur de propriété	Etiquette de propriété	Type	Valeur
OLC	C001	Octet string	Valeur de la structure H.245 OpenLogicalChannel. Réf.: Rec. UIT-T H.245
OLCack	C002	Octet string	Valeur de la structure H.245 OpenLogicalChannelAck. Réf.: Rec. UIT-T H.245
OLCcnf	C003	Octet string	Valeur de la structure H.245 OpenLogicalChannelConfirm. Réf.: Rec. UIT-T H.245
OLCrej	C004	Octet string	Valeur de la structure H.245 OpenLogicalChannelReject. Réf.: Rec. UIT-T H.245
CLC	C005	Octet string	Valeur de la structure H.245 CloseLogicalChannel. Réf.: Rec. UIT-T H.245
CLCack	C006	Octet string	Valeur de la structure H.245 CloseLogicalChannelAck. Réf.: Rec. UIT-T H.245

Annexe D

Transport sur protocole IP

D.1 Transport sur protocoles IP/UDP au moyen du verrouillage de niveau d'application

Les messages de protocole définis dans la présente Recommandation peuvent être transmis sur UDP. Si aucun port n'est fourni par le pair (se reporter au § 7.2.8), il convient d'envoyer les commandes au numéro de port par défaut: 2944 pour les opérations codées au format alphanumérique, ou 2945 pour celles codées au format binaire. Les réponses doivent être envoyées à l'adresse et au port d'origine des commandes correspondantes.

Le verrouillage ALF consiste en un ensemble de techniques qui permettent à une application, contrairement à un empilement, de décider de la manière dont seront envoyés les messages vers l'autre côté. Une technique de verrouillage ALF caractéristique est de permettre à une application de modifier l'ordre d'envoi des messages lorsque après la mise en file d'attente de ceux-ci, cette file s'allonge encore. Il n'existe aucune spécification formelle concernant le verrouillage ALF. Les procédures à l'Annexe D.1 contiennent un ensemble minimal de propositions de comportements de verrouillage ALF.

Il convient que les réalisateurs utilisant IP/UDP avec ALF connaissent les restrictions de MTU relatives à la taille maximale des messages.

D.1.1 Fourniture de la capacité fonctionnelle d'une fois au plus

Les messages transportés sur le protocole UDP peuvent subir des pertes. En l'absence de réponse opportune, les commandes sont répétées. La plupart des commandes ne sont pas idempotentes. L'état de la passerelle MG deviendra imprévisible si, par exemple, des commandes de type Add sont exécutées plusieurs fois. Les procédures de transmission doivent donc offrir une capacité d'une fois au plus.

Les entités de protocole homologues sont censées mémoriser une liste des réponses qu'elles ont envoyées à de récentes transactions ainsi qu'une liste des transactions qui restent en cours d'exécution. L'identificateur de transaction de chaque message entrant est comparé aux identificateurs de transaction des réponses récentes envoyées avec le même paramètre MId. S'il y a une correspondance, l'entité n'exécute pas la transaction mais répète simplement la réponse. S'il n'y aucune correspondance, le message est comparé à la liste des transactions couramment en attente. S'il y a une correspondance dans cette liste, indiquant une transaction dupliquée, l'entité n'exécute pas la transaction [se reporter au § D.1.4 pour les procédures concernant l'envoi de TransactionPending (Transaction en attente ou en cours)].

La procédure utilise une valeur de temporisation plus longue, appelée ci-après LONG-TIMER. La temporisation doit être réglée de façon à être supérieure à la durée maximale d'une transaction afin de tenir compte du nombre maximal de répétitions, de la valeur maximale du compteur de répétitions et du temps de propagation maximal d'un paquet dans le réseau. Une valeur de 30 secondes est suggérée.

La copie des réponses peut être détruite soit LONG-TIMER secondes après l'envoi de la réponse ou lorsque la passerelle de l'entité reçoit – au moyen du "paramètre d'acquittement de réponse" – confirmation du fait que la réponse a été reçue. Pour les transactions qui sont acquittées au moyen de ce paramètre, l'entité doit conserver une copie de l'identificateur de transaction pendant LONG-TIMER secondes après l'envoi de la réponse, afin de détecter et d'ignorer les copies miroirs de la demande de transaction qui pourraient être produites par le réseau.

D.1.2 Identificateurs de transaction et dialogue à trois

D.1.2.1 Identificateurs de transaction

Les identificateurs de transaction sont des nombres entiers de 32 bits. Un contrôleur MGC peut décider d'utiliser un espace numérique spécifique pour chacune des passerelles MG qu'il gère. Il peut également utiliser le même espace numérique pour toutes les passerelles MG qui appartiennent à un même groupe arbitraire. Les contrôleurs MGC peuvent décider de partager entre plusieurs processus indépendants la charge de gérer une grande passerelle MG. Ces processus partageront le même espace numérique de transaction. Il existe de multiples implémentations possibles de cette mise en commun, par exemple une attribution centralisée des identificateurs de transaction ou une préaffectation à différents processus d'étendues d'identificateurs ne se chevauchant pas. Les implémentations doivent garantir que des identificateurs de transaction uniques sont attribués à toutes les transactions issues d'un contrôleur MGC logique (mId identique). Les passerelles MG peuvent simplement détecter les transactions en double en verrouillant au niveau de l'identificateur de transaction et de l'identificateur mld seulement.

D.1.2.2 Dialogue à trois

Le paramètre d'acquittement de réponse de transaction peut se trouver dans un message quelconque. Ce paramètre contient le paquetage des "étendues d'identificateurs de transaction confirmés". Les entités peuvent choisir de supprimer les copies des réponses aux transactions dont l'identificateur est inclus dans les "étendues d'identificateurs de transaction confirmés" qui ont été reçues dans les messages de réponse concernant les transactions. Ces entités doivent ignorer sans notification les commandes ultérieures si l'identificateur de transaction ne se trouve pas dans ces étendues.

Les valeurs des "étendues d'identificateurs de transaction confirmés" ne doivent pas être utilisées si plus de LONG-TIMER secondes se sont écoulées depuis que la passerelle MG a envoyé sa dernière réponse au contrôleur MGC ou si une passerelle MG reprend son fonctionnement. Dans cette situation, il y a lieu d'accepter et de traiter les transactions sans effectuer de test sur l'identificateur de transaction.

Les messages qui acheminent le paramètre "Acquittement de réponse de transaction" peuvent être transmis dans un ordre quelconque. L'entité doit conserver "l'étendue d'identificateurs de transaction confirmés" reçue pendant LONG-TIMER secondes.

Dans le codage binaire, si seulement l'élément `firstAck` est présent dans l'acquittement d'une réponse (se reporter au § A.2), une seule transaction est acquittée. Si les deux éléments `firstAck` et `lastAck` sont présents, c'est l'étendue des transactions comprises entre le premier et le dernier acquittement qui est acquittée. Dans le codage alphanumérique, un trait horizontal est utilisé pour indiquer une plage de transactions en cours d'acquittement (se reporter au § B.2).

D.1.3 Calcul des temporisations de retransmission

Il appartient à l'entité émettrice de la demande de fournir des temporisations appropriées pour toutes les transactions en cours et de réessayer les transactions dont la temporisation a été dépassée. Par ailleurs, lorsque des transactions répétées n'obtiennent pas d'acquittement, il appartient à l'entité émettrice de la demande de rechercher des services redondants et de libérer des connexions actives ou en instance.

La spécification évite à dessein de prescrire de quelconques valeurs pour les temporisateurs de retransmission. Ces valeurs dépendent normalement du réseau. Il y a lieu que les temporisateurs de retransmission estiment normalement leur valeur en mesurant la durée écoulée entre l'envoi d'une commande et le retour d'une réponse. Les implémentations DOIVENT garantir que l'algorithme utilisé pour calculer la temporisation de retransmission effectue un recul de la temporisation de retransmission qui augmente exponentiellement pour chaque retransmission ou répétition après la première.

NOTE – Une possibilité consiste à utiliser l'algorithme implémenté dans le protocole TCP-IP, qui fait appel à deux variables comme suit:

- le temps d'acquittement moyen (AAD, *average acknowledgement delay*), qui est estimé par un calcul de moyenne lissée exponentiellement des temps observés;
- l'écart moyen (ADEV, *average deviation*) qui est estimé par un calcul de moyenne lissée exponentiellement de la valeur absolue de la différence entre le temps observé et la moyenne actuelle. Dans le protocole TCP, le temporisateur de retransmission est réglé à la somme du temps moyen plus N fois l'écart moyen. Il convient de limiter la valeur maximale du temporisateur pour le protocole défini dans la présente Recommandation afin de garantir qu'aucun paquet répété ne sera reçu par les passerelles après LONG-TIMER secondes. Une valeur maximale de 4 secondes est suggérée.

Après toute retransmission, l'entité DOIT effectuer les opérations suivantes:

- doubler la valeur estimée du temps d'acquittement moyen (AAD);
- calculer une valeur aléatoire, répartie uniformément entre 0,5 AAD et AAD;
- régler le temporisateur de retransmission sur la somme de cette valeur aléatoire et N fois l'écart moyen.

Cette procédure a deux effets. Comme elle comporte un composant qui augmente exponentiellement, elle ralentit automatiquement le flux de messages en cas d'encombrement. Comme elle comporte également un composant aléatoire, elle rompt l'éventuelle synchronisation entre les notifications déclenchées par le même événement externe.

D.1.4 Réponses provisoires

L'exécution de certaines transactions peut demander beaucoup de temps. De longs délais d'exécution peuvent entrer en interaction avec la procédure de retransmission temporisée et donner lieu soit à un nombre anormal de retransmissions ou à des valeurs de temporisation devenant trop longues pour être efficaces. Les entités qui peuvent prédire qu'une transaction nécessitera un long temps d'exécution peuvent envoyer une réponse provisoire, de type "transaction en cours". Elles DOIVENT envoyer cette réponse si elles reçoivent une répétition d'une transaction encore en cours d'exécution.

Les entités qui reçoivent un message de "transaction en cours" doivent commuter sur un temporisateur de répétition différent pour les transactions répétitives. La terminaison racine possède une propriété (ProvisionalResponseTimerValue) qui peut être réglée au nombre maximal demandé de millisecondes entre la réception d'une commande et l'émission de la réponse "transaction en cours". A la réception d'une réponse finale suivant la réception de réponses provisoires, une confirmation immédiate doit être envoyée et les temporisateurs de répétition normaux envoyés par la suite. Une entité qui envoie une réponse provisoire DOIT inclure le champ immAckRequired dans la réponse finale qui suit en indiquant qu'une confirmation immédiate est attendue. La réception d'une transaction en attente après une réponse doit être ignorée.

D.1.5 Répétitions des demandes, réponses et acquittements

Le protocole est organisé sous la forme d'un paquetage de transactions dont chacune se compose d'une demande et d'une réponse couramment appelée acquittement**.** Les messages du protocole, acheminés sur UDP, peuvent subir des pertes. En l'absence de réponse opportune, les transactions sont répétées. Les entités sont censées mémoriser une liste des réponses qu'elles ont envoyées à des transactions récentes, c'est-à-dire une liste de toutes les réponses qu'elles ont envoyées au cours des dernières LONG-TIMER secondes, ainsi qu'une liste des transactions qui sont en cours d'exécution.

Le mécanisme de répétition est utilisé pour éviter trois types d'erreur possibles:

- les erreurs de transmission, lorsque par exemple un paquet est perdu à cause du bruit en ligne ou d'un encombrement dans une file d'attente;
- les défaillances de composants, lorsque par exemple une interface avec une entité devient indisponible;
- les défaillance d'entité, lorsque par exemple une entité entière devient indisponible.

Les entités doivent être en mesure de déduire, de l'historique enregistré, une estimation du taux de pertes de paquets en raison d'erreurs de transmission. Dans un système correctement configuré, ce taux de pertes doit être maintenu à une valeur très basse, normalement inférieure à 1%. Si un contrôleur MGC ou une passerelle MG doit répéter un message plus d'un certain nombre de fois, il est tout à fait normal de partir du principe qu'il se produit autre chose qu'une erreur de transmission. Par exemple, pour un taux de pertes de 1%, la probabilité que cinq tentatives de transmission consécutives échouent est de 10^{-11}, soit un événement qui devrait se produire moins souvent que tous les 10 jours pour une passerelle MG traitant 1000 transactions par seconde. (En fait, le nombre de répétitions qui est considéré comme excessif devrait être fonction du taux de pertes de paquets observé.) Il convient de remarquer que le "seuil de suspicion", que l'on appellera "Max1" est normalement inférieur au "seuil de déconnexion", qui devrait être fixé à une valeur beaucoup plus grande.

Un algorithme classique de retransmission comptera simplement le nombre de répétitions successives et conclura que l'association est rompue après la réémission du même paquet un nombre de fois excessif (normalement entre 7 et 11 fois). Pour tenir compte de la possibilité d'une reprise sur défaillance non détectée ou en cours, l'on modifie l'algorithme classique de façon que, si la passerelle MG reçoit un message ServiceChange valide, annonçant une reprise sur défaillance, cette

passerelle commence à envoyer les commandes en instance à ce nouveau contrôleur MGC. Les réponses aux commandes continueront d'être envoyées à l'adresse source de chaque commande.

Afin d'assurer une adaptation automatique à la charge du réseau, la présente Recommandation spécifie des temporisateurs à croissance exponentielle. Si la temporisation initiale est réglée à 200 ms, la perte d'une cinquième retransmission sera détectée au bout d'environ 6 secondes. C'est sans doute un temps d'attente acceptable pour détecter une reprise sur défaillance. Les répétitions devront continuer après ce délai, non seulement pour résoudre un éventuel problème de connexité transitoire mais une attente totale de 30 secondes est sans doute acceptable aussi pour donner un peu plus de temps pour l'exécution d'une reprise sur défaillance.

Il importe cependant que le temps maximal alloué aux retransmissions soit limité. Avant toute retransmission, l'on vérifie que le temps écoulé depuis l'envoi du datagramme initial n'est pas plus grand que T-MAX. Si une durée supérieure à T-MAX s'est écoulée, la passerelle MG en conclut que le contrôleur MGC a subi une défaillance et elle commence son processus de reprise comme décrit au § 11.5. La passerelle MG doit utiliser une commande ServiceChange avec le paramètre ServiceChangeMethod réglé sur Disconnected de façon que le nouveau contrôleur MGC sache que la passerelle MG a perdu une ou plusieurs transactions. La valeur T-MAX est associée à la valeur LONG-TIMER, qui est obtenue par addition à T-MAX du temps maximal de propagation dans le réseau.

D.2 Utilisation de TCP

Les messages de protocoles définis dans la présente Recommandation peuvent être transmis sur le protocole TCP. Si aucun accès n'est spécifié par l'autre côté (se reporter au § 7.2.8), les commandes doivent être envoyées à l'accès par défaut. Le protocole a des messages pour unité de transfert alors que le protocole TCP est un protocole orienté vers le flux. TPKT, conformément à RFC 1006, DOIT être utilisé pour détailler les messages à l'intérieur du flux TCP.

Dans un protocole orienté vers les transactions, il reste possible que des demandes ou des réponses transactionnelles soient perdues. A ce propos, il est recommandé que les entités utilisant le transport TCP implémentent des temporisateurs de niveau d'application pour chaque demande et chaque réponse, comme ceux qui sont spécifiés pour le verrouillage des niveaux d'application en protocole UDP.

D.2.1 Fourniture de la capacité fonctionnelle d'une fois au plus

Les messages acheminés par TCP ne sont pas exposés aux pertes de transport mais, dans les implémentations concrètes, on peut noter des pertes de demande ou de réponse transactionnelle. En l'absence de réponse opportune, les commandes sont répétées. La plupart des commandes ne sont pas idempotentes. L'état de la passerelle MG deviendra imprévisible si, par exemple, des commandes de type Add sont exécutées plusieurs fois.

Afin d'éviter de telles pertes, il est recommandé que les entités suivent les procédures du § D.1.1.

D.2.2 Identificateurs de transaction et dialogue à trois

Pour les mêmes raisons, il est possible que des réponses de transaction effectuée soient perdues malgré un protocole d'acheminement fiable comme le TCP. Il est recommandé que les entités suivent les procédures du § D.1.2.2.

D.2.3 Calcul des temporisations de retransmission

Avec un acheminement fiable, l'incidence des pertes de demandes/réponses de transaction est censée être très faible. De simples mécanismes de temporisation seront donc suffisants. Des algorithmes de ralentissement exponentiel ne devraient pas être nécessaires bien qu'ils puissent être employés lorsque, dans un contrôleur MGC par exemple, le code correspondant est déjà requis car les contrôleurs MGC doivent implémenter le protocole ALF/UDP en plus du protocole TCP.

D.2.4 Réponses provisoires

Comme avec le protocole UDP, l'exécution de certaines transactions peut demander beaucoup de temps. Les entités qui peuvent prédire qu'une transaction nécessitera un long temps d'exécution peuvent envoyer une réponse provisoire, de type "transaction en cours". Elles doivent envoyer cette réponse si elles reçoivent une répétition d'une transaction qui est encore en cours d'exécution.

Les entités qui reçoivent un message de "transaction en cours" doivent commuter sur un temporisateur de répétition plus long pour cette transaction.

Les entités doivent conserver les transactions et les réponses jusqu'à ce qu'elles aient été confirmées. La procédure de base du § D.1.4 devrait être suivie mais de simples valeurs de temporisation devraient suffire. Il n'est pas nécessaire d'envoyer une confirmation immédiate à la réception d'une réponse finale.

D.2.5 Ordonnancement des commandes

Le protocole TCP assure un acheminement ordonné des transactions. Aucune procédure spéciale n'est requise. Il convient de noter que le protocole ALF/UDP permet à une entité expéditrice de modifier son comportement en cas d'encombrement. Il permet en particulier à cette entité de réordonner les transactions lorsqu'un encombrement est détecté. Le protocole TCP ne peut pas donner les mêmes résultats.

Annexe E

Paquetages de base

La présente annexe contient les définitions de quelques paquetages à utiliser dans le cadre de la Rec. UIT-T H.248.1.

E.1 Paquetage générique

Identificateur de paquetage: g (0x0001)

Version: 1

Extension: aucune

Description:

paquetage générique pour les items communément rencontrés

E.1.1 Propriétés

Aucune

E.1.2 Evénements

Cause

Identificateur d'événement: cause (0x0001)

Evénement d'erreur générique

Paramètres EventsDescriptor (descripteur d'événements) : aucun

Paramètres de descripteur ObservedEvents (événements observés):

Cause d'ordre général

Identificateur de paramètre: Generalcause (0x0001)

Ce paramètre répartit les défaillances en six groupes sur lesquels le contrôleur MGC peut agir.

Type de paramètre: énumération

Valeurs possibles:

"NR" (*normal release*) libération normale (0x0001)

"UR" (*unavailable resources*) ressources non disponibles (0x0002)

"FT" (*failure, temporary*) défaillance, temporaire (0x0003)

"FP" (*failure, permanent*) défaillance, permanente (0x0004)

"IW" (*interworking error*) erreur d'interfonctionnement (0x0005)

"UN" (*unsupported*) non pris en charge (0x0006)

Cause de la défaillance

Identificateur de paramètre: Failurecause (0x0002)

Valeurs possibles: OCTET STRING

Description: la cause de la défaillance est la valeur générée par l'équipement libéré, c'est-à-dire une connexion réseau libérée. La valeur concernée est définie dans le protocole approprié de commande de support.

Achèvement de signal

Identificateur d'événement: sc (0x0002)

Indique la fin d'un signal pour lequel le paramètre notifyCompletion était réglé de manière à permettre le rapport d'un événement d'achèvement. Pour une description procédurale plus poussée, se reporter aux § 7.1.1, 7.1.17 et 7.2.7.

Paramètres EventsDescriptor (descripteur d'événements) : aucun

Paramètres de descripteur ObservedEvents (événements observés):

Identité du signal

Identificateur de paramètre: SigID (0x0001)

Ce paramètre identifie le signal qui s'est achevé. Lorsqu'un signal est contenu dans une liste de signaux, le paramètre d'identification de cette liste, indiquant la liste appropriée, devrait aussi être renvoyé.

Type de paramètre: binaire: octet (string), texte: string

Valeurs possibles: un signal qui s'est achevé. Un signal doit être identifié à l'aide de la syntaxe de pkgdName sans utiliser de remplacement par des caractères génériques.

Méthode de terminaison

Identificateur de paramètre: Meth (0x0002)

Indique le moyen par lequel le signal s'est achevé.

Type de paramètre: énumération

Valeurs possibles:

"TO" (0x0001) signal interrompu ou s'étant achevé autrement de lui-même

"EV" (0x0002) interrompu par un événement

"SD" (0x0003) arrêté par un nouveau descripteur Signals

"NC" (0x0004) inachevé, pour une autre raison

Identificateur de la liste de signaux

Identificateur de paramètre: SLID (0x0003)

Indique la liste de signaux à laquelle un signal appartient. L'identificateur de la liste de signaux n'est renvoyé que dans les cas où le signal figure sur une liste de signaux.

Type: entier

Valeurs possibles: tout entier

E.1.3 Signaux

Aucun.

E.1.4 Statistiques

Aucune.

E.2 Paquetage racine de base

Propriétés paquetage racine de base

Identificateur de paquetage: root (0x0002)

Version: 2

Extension: aucune

Description:

ce paquetage définit des propriétés larges de passerelle.

E.2.1 Propriétés

MaxNrOfContexts

Identificateur de propriété: maxNumberOfContexts (0x0001)

La valeur de cette propriété donne le nombre maximal de contextes qui peuvent exister à n'importe quel moment. Le contexte NULL (néant) n'est pas compris dans ce nombre.

Type de paramètre: double

Valeurs possibles: 1 et plus

Défini dans: TerminationState

Caractéristiques: lecture seulement

MaxTerminationsPerContext

Identificateur de propriété: maxTerminationsPerContext (0x0002)

Il indique le nombre maximal de terminaisons autorisées dans un contexte (se reporter au § 6.1)

Type de paramètre: nombre entier

Valeurs possibles: n'importe quel entier

Défini dans: TerminationState

Caractéristiques: lecture seulement

normalMGExecutionTime

PropertyId: normalMGExecutionTime (0x0003)

Réglable par le contrôleur MGC afin d'indiquer l'intervalle pendant lequel le contrôleur MGC doit attendre de la passerelle MG une réponse à une transaction quelconque (à l'exclusion du retard réseau)

Type de paramètre: Integer (entier)

Valeurs possibles: n'importe quel entier, indique des millisecondes

Défini dans : TerminationState

Caractéristiques: lecture/écriture

normalMGCExecutionTime

PropertyId: normalMGCExecutionTime (0x0004)

Réglable par le contrôleur MGC afin d'indiquer l'intervalle pendant lequel la passerelle MG doit attendre du contrôleur MGC une réponse à une transaction quelconque (à l'exclusion du retard réseau)

Type de paramètre: Integer (entier)

Valeurs possibles: n'importe quel entier, indique des millisecondes

Défini dans : TerminationState

Caractéristiques: lecture/écriture

MGProvisionalResponseTimerValue

PropertyId: MGProvisionalResponseTimerValue (0x0005)

Cette propriété indique le temps pendant lequel le contrôleur MGC devrait attendre une réponse en souffrance de la passerelle MG si une transaction ne peut pas s'achever. Elle est réglée initialement à normalMGExecutionTime, plus le retard réseau, mais elle peut être abaissée.

Type de paramètre: Integer (entier)

Valeurs possibles: n'importe quel entier, indique des millisecondes

Défini dans: TerminationState

Caractéristiques: lecture/écriture

MGCProvisionalResponseTimerValue

PropertyId: MGCProvisionalResponseTimerValue (0x0006)

Cette propriété indique le temps pendant lequel la passerelle MG devrait attendre une réponse en souffrance du contrôleur MGC si une transaction ne peut pas s'achever. Elle est réglée initialement à normalMGCExecutionTime, plus le retard réseau, mais elle peut être abaissée.

Type de paramètre: Integer (entier)

Valeurs possibles: n'importe quel entier, indique des millisecondes

Défini dans: TerminationState

Caractéristiques: lecture/écriture

MGCOriginatedPendingLimit

PropertyId: MGCOriginatedPendingLimit (0x0007)

Cette propriété indique le nombre de transaction en cours qui peuvent être reçues en provenance du contrôleur MGC. Lorsque cette limite est dépassée, le contrôleur MGC devrait émettre une réponse TransactionReply avec l'erreur 506 (Nombre de transactions en cours dépassé), sinon la passerelle MG peut penser que la transaction est fausse.

Type de paramètre: Integer (entier)

Valeurs possibles: n'importe quel entier

Défini dans: TerminationState

Caractéristiques: lecture/écriture

MGOriginatedPendingLimit

PropertyId: MGOriginatedPendingLimit (0x0008)

Cette propriété indique le nombre de transaction en cours qui peuvent être reçues en provenance de la passerelle MG. Lorsque cette limite est dépassée, la passerelle MG devrait émettre une réponse TransactionReply avec l'erreur 506 (Nombre de transactions en cours dépassé), sinon le contrôleur MGC peut penser que la transaction est fausse.

Type de paramètre: Integer (entier)

Valeurs possibles: n'importe quel entier

Défini dans: TerminationState

Caractéristiques: lecture/écriture

E.2.2 Evénements

Aucun.

E.2.3 Signaux

Aucun.

E.2.4 Statistiques

Aucune.

E.2.5 Procédures

Aucune.

E.3 Paquetage de générateur de tonalités

Identificateur de paquetage: tonegen (0x0003)

Version: 1

Extension: aucune

Description:

ce paquetage définit des signaux afin de produire des tonalités audio. Il ne spécifie pas de valeurs paramétriques. Il est destiné à être extensible. Généralement, les tonalités sont définies comme un signal individuel assorti d'un paramètre, ind, représentant le délai entre chiffres et un identificateur de tonalité à utiliser avec les tonalités restituables. La compatibilité de chaque identificateur de tonalité doit être conservée avec toute production de la même tonalité. Les passerelles MG sont censées être configurées avec les caractéristiques des tonalités appropriées au pays dans lequel ces passerelles sont situées.

Conçu pour être étendu seulement: oui

E.3.1 Propriétés

Aucune.

E.3.2 Evénements

Aucun.

E.3.3 Signaux

Tonalité exécutable

Identificateur de signal: pt (0x0001)

Restitution d'une tonalité audio dans une voie audio

Type de signal: bref

Durée: fournie

Paramètres additionnels:

Liste d'identificateurs de tonalité

Identificateur de paramètre: tl (0x0001)

Type de paramètre: liste d'identificateurs de tonalité

Liste de tonalités à restituer en séquence. Cette liste DOIT contenir un ou plusieurs identificateurs de tonalité.

Durée entre signaux

Identificateur de paramètre: ind (0x0002)

Type de paramètre: entier

Temporisation entre deux tonalités consécutives, en millisecondes.

Aucun identificateur de tonalité n'est spécifié dans ce paquetage. Les paquetages qui étendent celui-ci peuvent ajouter d'autres valeurs d'identificateur de tonalité et d'autres signaux de tonalité individuelle.

E.3.4 Statistiques

Aucune.

E.3.5 Procédures

Aucune.

E.4 Paquetage de détection de tonalité

Identificateur de paquetage: tonedet (0x0004)

Version: 1

Conçu pour être étendu seulement: oui

Extension: aucune

Ce paquetage définit des événements pour la détection de tonalités audio. Celles-ci sont choisies par leur nom (identificateur de tonalité). Les passerelles MG sont censées être configurées avec les caractéristiques des tonalités appropriées au pays dans lequel ces passerelles sont situées.

Ce paquetage ne spécifie pas de valeurs paramétriques. Il est destiné à être extensible.

E.4.1 Propriétés

Aucune.

E.4.2 Evénements

Détection de début de tonalité

Identificateur d'événement: std, 0x0001

Détecte le début d'une tonalité. Les caractéristiques d'une détection positive de tonalité dépendent de l'implémentation.

Paramètres de descripteur d'événements:

liste d'identificateurs de tonalité

identificateur de paramètre: tl (0x0001)

type de paramètre: liste d'identificateurs de tonalité

valeurs possibles: le seul identificateur de tonalité défini dans ce paquetage est "caractère générique", qui est l'astérisque "*" en codage alphanumérique et qui est 0x0000 en codage binaire. Les extensions de ce paquetage ajouteront des valeurs possibles d'identificateur de tonalité. Si le paramètre tl a la valeur "caractère générique", tout identificateur de tonalité sera détecté.

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor):

identificateur de tonalité

identificateur de paramètre: tid (0x0003)

type de paramètre: énumération

valeurs possibles: la valeur "caractère générique" définie ci-dessus est la seule valeur définie dans ce paquetage. Les extensions de ce paquetage ajouteront d'autres valeurs possibles d'identificateur de tonalité.

Détection de fin de tonalité

Identificateur d'événement: etd, 0x0002

Détecte la fin d'une tonalité.

Paramètres de descripteur d'événements:

liste d'identificateurs de tonalité

identificateur de paramètre: tl (0x0001)

type de paramètre: énumération ou liste de types énumérés

valeurs possibles: aucune valeur possible n'est spécifiée dans ce paquetage. Des extensions de ce paquetage ajouteront des valeurs possibles d'identificateur de tonalité.

Paramètres du descripteur ObservedEventsDescriptor:

identificateur de tonalité

identificateur de paramètre: tid (0x0003)

type de paramètre: énumération

valeurs possibles: la valeur "caractère générique" définie ci-dessus est la seule valeur définie dans ce paquetage. Des extensions de ce paquetage ajouteront des valeurs possibles d'identificateur de tonalité.

Durée

Identificateur de paramètre: dur (0x0002)

Type de paramètre: entier, en millisecondes

Ce paramètre contient la durée de la tonalité, de la première détection jusqu'à l'arrêt.

Détection de tonalité longue

Identificateur d'événement: ltd, 0x0003

Détecte qu'une tonalité est restituée pendant au moins une certaine durée

Paramètres de descripteur d'événements:

liste d'identificateurs de tonalité

identificateur de paramètre: tl (0x0001)

type de paramètre: énumération ou liste

valeurs possibles: la valeur "caractère générique" définie ci-dessus est la seule valeur définie dans ce paquetage. Des extensions de ce paquetage ajouteront des valeurs possibles d'identificateur de tonalité.

Durée:

identificateur de paramètre: dur (0x0002)

type de paramètre: entier, durée de référence pour l'essai

valeurs possibles: tout entier autorisé, exprimé en millisecondes

Paramètres du descripteur ObservedEventsDescriptor:

identificateur de tonalité:

identificateur de paramètre: tid (0x0003)

type: énumération

valeurs possibles: aucune valeur possible n'est spécifiée dans ce paquetage. Des extensions de ce paquetage ajouteront des valeurs possibles d'identificateur de tonalité.

E.4.3 Signaux

Aucun.

E.4.4 Statistiques

Aucune.

E.4.5 Procédures

Aucune.

E.5 Paquetage de générateur de tonalités DTMF de base

Identificateur de paquetage: dg (0x0005)

Version: 1

Extension: tonegen version 1

Ce paquetage définit les tonalités DTMF de base sous la forme de signaux et constitue une extension des valeurs autorisées du paramètre tl de tonalités restituables dans le paquetage tonegen.

E.5.1 Propriétés

Aucune.

E.5.2 Evénements

Aucun.

E.5.3 Signaux

Caractère DTMF 0

Identificateur de signal: d0 (0x0010)

Produit une tonalité DTMF 0 dont les caractéristiques sont définies dans la passerelle.

Type de signal: Brief (bref)

Durée: fournie

Paramètres additionnels:

aucun

Valeurs additionnelles:

d0 (0x0010) est défini comme identificateur de tonalité restituable

Les autres caractères DTMF sont spécifiés exactement de la même façon. Un tableau avec tous les noms de signaux et tous les identificateurs de signaux est inclus. Il convient de noter que chaque caractère DTMF est défini à la fois comme signal et comme identificateur de tonalité (tone id), ce qui étend le paquetage de production de tonalités de base. Il convient également de noter que les identificateurs de signaux DTMF sont différents des noms utilisés dans un script de numérotation.

Nom de signal	Ident. de signal/ident. de tonalité
Caractère DTMF 0	d0 (0x0010)
Caractère DTMF 1	d1 (0x0011)
Caractère DTMF 2	d2 (0x0012)
Caractère DTMF 3	d3 (0x0013)
Caractère DTMF 4	d4 (0x0014)
Caractère DTMF 5	d5 (0x0015)
Caractère DTMF 6	d6 (0x0016)
Caractère DTMF 7	d7 (0x0017)
Caractère DTMF 8	d8 (0x0018)
Caractère DTMF 9	d9 (0x0019)
Caractère DTMF *	ds (0x0020)
Caractère DTMF #	do (0x0021)
Caractère DTMF A	da (0x001a)
Caractère DTMF B	db (0x001b)
Caractère DTMF C	dc (0x001c)
Caractère DTMF D	dd (0x001d)

E.5.4 Statistiques

Aucune.

E.5.5 Procédures

Aucune.

E.6 Paquetage de détection de tonalités DTMF

Identificateur de paquetage: dd (0x0006)

Version: 1

Extension: tonedet version 1

Ce paquetage définit la détection des tonalités DTMF de base. Il constitue une extension des valeurs possibles d'identificateur de tonalité contenues dans les événements "détection de début de tonalité", "détection de fin de tonalité" et "détection de tonalité longue".

Les valeurs additionnelles d'identificateur de tonalité sont tous les identificateurs de tonalité décrits dans le paquetage dg (générateur de tonalités DTMF de base).

Le tableau ci-après mappe les événements DTMF aux symboles du script de numérotation comme il est décrit au § 7.1.14.

DTMF	Symbole d'événement
d0	"0"
d1	"1"
d2	"2"
d3	"3"
d4	"4"
d5	"5"
d6	"6"
d7	"7"
d8	"8"
d9	"9"
da	"A" ou "a"
db	"B" ou "b"
dc	"C" ou "c"
dd	"D" ou "d"
ds	"E" ou "e"
do	"F" ou "f"

E.6.1 Propriétés

Aucune.

E.6.2 Evénements

Chiffres DTMF

Les identificateurs d'événements (EventId) sont définis avec les mêmes noms que les identificateurs de signaux (SignalId) définis dans le tableau du § E.5.3.

Evénement d'achèvement de script de numérotation (DigitMap Completion Event)

Identificateur d'événement: ce, 0x0004

Il est généré lorsqu'un script de numérotation s'achève comme il est décrit dans le § 7.1.14.

Paramètres de descripteur d'événements: aucun.

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor):

DigitString

Identificateur de paramètre: ds (0x0001)

Type de paramètre: chaîne de symboles de script de numérotation (éventuellement vide) retournée comme quotedString

Valeurs possibles: séquence des caractères "0" à "9", "A" à "F", et le modificateur de longue durée "Z".

Description: partie de la chaîne de numérotation actuelle comme décrit au § 7.1.14 qui correspond à tout ou partie de la séquence d'événement en variante spécifiée dans le script de numérotation.

Méthode de terminaison

Identificateur de paramètre: Meth (0x0003)

Type de paramètre: énumération

Valeurs possibles:

"UM" (0x0001) correspondance non ambiguë

"PM" (0x0002) correspondance partielle, achèvement par l'expiration de la temporisation ou par un événement sans correspondance

"FM" (0x0003) correspondance totale, achèvement par l'expiration de la temporisation ou par un événement sans correspondance

Description: indique la raison de la création de l'événement. Se reporter aux procédures dans le § 7.1.14.

E.6.3 Signaux

Aucun.

E.6.4 Statistiques

Aucune.

E.6.5 Procédures

Le script de numérotation est activé uniquement si un descripteur d'événements est activé, qui contient un événement de fin de script de numérotation, comme défini au § E.6.2, et si cet événement de fin de script de numérotation contient un champ eventDM dans les actions requises, tel que défini au § 7.1.9. D'autres paramètres tels que KeepActive ou des événements intégrés de descripteurs de signaux peuvent aussi être présents dans le descripteur d'événements, sans affecter l'activation du script de numérotation.

E.7 Paquetage de générateur de tonalités de progression d'appel

Identificateur de paquetage: cg, 0x0007

Version: 1

Extension: tonegen version 1

Ce paquetage définit les tonalités de progression d'appel en tant que signaux et constitue une extension des valeurs autorisées du paramètre tl de tonalités restituables dans le paquetage tonegen.

E.7.1 Propriétés

Aucune.

E.7.2 Evénements

Aucun.

E.7.3 Signaux

Tonalité d'invitation

Identificateur de signal: dt (0x0030)

Produit une tonalité d'invitation à numéroter dont les caractéristiques sont disponibles dans la passerelle.

Type de signal: temporisé

Durée: fournie

Paramètres additionnels:

aucun

Valeurs additionnelles:

dt (0x0030) est défini comme identificateur des tonalités restituables

Les autres tonalités de ce paquetage sont définies exactement de la même façon. Un tableau avec tous les noms de signaux et tous les identificateurs de signaux est inclus. Il convient de noter que chaque tonalité est définie à la fois comme signal et comme identificateur de tonalité (tone id), ce qui étend le paquetage de production de tonalités de base.

Nom de signal	Ident. de signal/ident. de tonalité
Tonalité d'invitation	dt (0x0030)
Tonalité de retour d'appel	rt (0x0031)
Tonalité d'occupation	bt (0x0032)
Tonalité d'encombrement	ct (0x0033)
Tonalité spéciale d'information	sit (0x0034)
Tonalité d'avertissement (d'enregistrement)	wt (0x0035)
Tonalité d'identification de publiphone	prt (0x0036)
Tonalité d'appel en attente	cw (0x0037)
Tonalité d'attente de l'appelant	cr (0x0038)

E.7.4 Statistiques

Aucune.

E.7.5 Procédures

NOTE – L'ensemble requis d'identificateurs de tonalité correspond à celui qui est défini dans la Rec. UIT-T E.180/Q.35. L'explication de ces tonalités est donnée dans la Rec. UIT-T E.180/Q.35.

E.8 Paquetage de détection de tonalités de progression d'appel

Identificateur de paquetage: cd (0x0008)

Version: 1

Extension: tonedet version 1

Ce paquetage définit les tonalités de détection de progression d'appel de base. Il constitue une extension des valeurs possibles d'identificateur de tonalité contenues dans les événements "détection de début de tonalité", "détection de fin de tonalité" et "détection de tonalité longue".

Valeurs additionnelles

Les valeurs d'identificateur de tonalités définies pour la détection de début de tonalité, la détection de fin de tonalité et la détection de tonalité longue avec les mêmes que celles du paquetage cg (production de tonalités de progression d'appel).

L'ensemble requis d'identificateurs de tonalité correspond à celui qui est défini dans la Rec. UIT-T E.180/Q.35. L'explication de ces tonalités est donnée dans la Rec. UIT-T E.180/Q.35.

E.8.1 Propriétés

Aucune.

E.8.2 Evénements

Les événements sont définis comme dans le paquetage de générateur de tonalités de progression d'appel (cg) pour les tonalités qui sont énumérées dans le tableau du § E.7.3.

E.8.3 Signaux

Aucun.

E.8.4 Statistiques

Aucune.

E.8.5 Procédures

Aucune.

E.9 Paquetage de supervision de ligne analogique

Identificateur de paquetage: al, 0x0009

Version: 1

Extension: aucune

Ce paquetage définit les événements et les signaux pour une ligne analogique.

E.9.1 Propriétés

Aucune.

E.9.2 Evénements

Raccrochage

Identificateur d'événement: on (0x0004)

Détecte le raccrochage du combiné. Toutes les fois qu'un descripteur d'événements est activé pour demander une surveillance afin de détecter un événement de raccrochage et que la ligne est déjà raccrochée, la passerelle MG doit se comporter conformément au réglage du paramètre "strict".

Paramètres de descripteur d'événement

Transition stricte

Identificateur de paramètre: strict (0x0001)

Type de paramètre: énumération

Valeurs possibles: "exact" (0x00), "state" (0x01), "failWrong" (0x02)

"exact" signifie que seule une transition effective à l'état de raccrochage "raccroché" doit être reconnue;

"state" signifie que l'événement doit être reconnu si la transition à l'état de raccrochage est détectée ou si l'état de raccrochage est déjà "raccroché";

"failWrong" signifie que si l'état de raccrochage est déjà "raccroché", la commande échoue et une erreur est renvoyée.

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor)

Etat initial

Identificateur de paramètre: init (0x0002)

Type de paramètre: booléen

Valeurs possibles:

"True" (Vrai) signifie que l'événement a été signalé parce que la ligne était déjà raccrochée lorsque le descripteur d'événement contenant cet événement a été activé;

"False" (Faux) signifie que l'événement représente une transition effective à l'état "raccroché".

Décrochage

Identificateur d'événement: of (0x0005)

Détecte le décrochage du combiné. Toutes les fois qu'un descripteur d'événements est activé pour demander une surveillance afin de détecter un événement de décrochage et que la ligne est déjà décrochée, la passerelle MG doit se comporter conformément au réglage du paramètre "strict".

Paramètres de descripteur d'événement

Transition stricte

Identificateur de paramètre: strict (0x0001)

Type de paramètre: énumération

Valeurs possibles: "exact" (0x00), "state" (0x01), "failWrong" (0x02)

"exact" signifie que seule une transition effective à l'état de raccrochage "décroché" doit être reconnue;

"state" signifie que l'événement doit être reconnu si la transition à l'état de raccrochage est détectée ou si l'état de raccrochage est déjà "décroché";

"failWrong" signifie que si l'état de raccrochage est déjà "décroché", la commande échoue et une erreur est renvoyée.

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor)

Etat initial

Identificateur de paramètre: init (0x0002)

Type de paramètre: booléen

Valeurs possibles:

"True" (Vrai) signifie que l'événement a été signalé parce que la ligne était déjà décrochée lorsque le descripteur d'événement contenant cet événement a été activé;

"False" (Faux) signifie que l'événement représente une transition effective à l'état "décroché".

Manipulation de crochet

Identificateur d'événement: fl, 0x0006

Détecte la manipulation de crochet. Une manipulation de crochet correspond à un raccrochage suivi d'un décrochage dans un délai compris entre une durée minimale et une durée maximale.

Paramètre de descripteur d'événement

Durée minimale

Identificateur de paramètre: mindur (0x0004)

Type de paramètre: entier en millisecondes

Valeur par défaut fournie.

Durée maximale

Identificateur de paramètre: maxdur (0x0005)

Type de paramètre: entier en millisecondes

Valeur par défaut fournie.

Paramètres de descripteur d'événements observés

Aucun

E.9.3 Signaux

Sonnerie

Identificateur de signal: ri, 0x0002

S'applique au retour d'appel sur la ligne

Type du signal: temporisé

Durée: fournie

Paramètres additionnels:

cadence

Identificateur de paramètre: cad (0x0006)

Type de paramètre: liste d'entiers représentant des durées de segments commutés alternativement, qui constituent un cycle de sonnerie complet à partir d'une commutation active. Unités: millisecondes

La valeur par défaut est fixe ou configurée. Les passerelles MG à fonctions restreintes peuvent ne pas tenir compte des valeurs de cadence qu'elles ne sont pas en mesure de produire.

fréquence

Identificateur de paramètre: freq (0x0007)

Type de paramètre: entier en Hz

La valeur par défaut est fixe ou configurée. Les passerelles MG à fonctions restreintes peuvent ne pas tenir compte des valeurs de cadence qu'elles ne sont pas en mesure de produire.

E.9.4 Statistiques

Aucune.

E.9.5 Procédures

Si le contrôleur MGC définit un paramètre EventsDescriptor contenant un événement de transition d'état de raccrochage (raccroché ou décroché) avec le paramètre "strict" (0x0001) réglé sur "failWrong", et que l'état de raccrochage correspond déjà à ce que la transition implique, l'exécution de la commande contenant le paramètre EventsDescriptor échoue. La passerelle MG DOIT inclure dans sa réponse le code d'erreur 540 (état de raccrochage initial inattendu).

E.9.6 Code d'erreur

Ce paquetage définit un nouveau code d'erreur:

540 (état de raccrochage initial inattendu)

La procédure d'utilisation de ce code est détaillée dans § E.9.5.

E.10 Paquetage de continuité de base

Identificateur de paquetage: ct (0x000a)

Version: 1

Extension: aucune

Ce paquetage définit des événements et des signaux pour essais de continuité. L'essai de continuité comprend la fourniture d'un bouclage ou d'une fonctionnalité d'émetteur-récepteur.

E.10.1 Propriétés

Aucune.

E.10.2 Evénements

Exécution

Identificateur d'événement: cmp, 0x0005

Cet événement détecte une fin d'essai pour l'essai de continuité.

Paramètres de descripteur d'événement

Aucun

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor)

Résultat

Identificateur de paramètre: res (0x0008)

Type de paramètre: énumération

Valeurs possibles: succès (0x0001), échec (0x0000)

E.10.3 Signaux

Essai de continuité

Identificateur de signal: ct (0x0003)

Déclenche l'envoi d'une tonalité d'essai de continuité sur la terminaison à laquelle il est appliqué.

Type du signal: temporisé

Valeur par défaut configurée

Paramètres additionnels:

aucun

Réponse

Identificateur de signal: rsp (0x0004)

Ce signal est utilisé pour répondre à un essai de continuité. Se reporter au § E.10.5 pour une explication plus poussée.

Type du signal: commuté

Une durée par défaut configurée

Paramètres additionnels:

aucun

E.10.4 Statistiques

Aucune.

E.10.5 Procédures

Lorsqu'il veut déclencher un essai de continuité, un contrôleur MGC envoie à la passerelle MG une commande qui contient:

- un descripteur de signaux avec le signal ct;
- un descripteur d'événements contenant l'événement cmp.

A la réception d'une commande contenant le signal ct et l'événement cmp, la passerelle MG déclenche la tonalité d'essai de continuité pour la terminaison spécifiée. Si la tonalité de retour est détectée et que toutes les autres conditions requises sont remplies avant l'expiration de la temporisation du signal, l'événement cmp doit être créé avec la valeur du paramètre de résultat égale à "success" (succès). Dans tous les autres cas, l'événement cmp doit être créé avec la valeur du paramètre de résultat égale à "failure" (échec).

Lorsqu'un contrôleur MGC veut que la passerelle MG réponde à un essai de continuité, il envoie à cette passerelle une commande qui contient un descripteur de signal avec le signal rsp. A la réception d'une commande avec le signal rsp, la passerelle MG applique soit un bouclage ou (pour les circuits à 2 fils) attend la réception d'une tonalité d'essai de continuité. Dans le cas du bouclage, toute information entrante sera renvoyée comme information sortante. Dans le cas du circuit à 2 fils, chaque fois que la tonalité d'essai appropriée est reçue, la tonalité de réponse appropriée doit être envoyée. Le contrôleur MGC détermine quand supprimer le signal rsp.

Lorsqu'un essai de continuité est exécuté sur une terminaison, aucun dispositif d'écho ou codec ne doit être actif sur cette terminaison.

L'exécution d'une assurance de conduit vocal en tant que partie d'un essai de continuité est configurée par accord bilatéral entre les exploitants de réseau.

NOTE – Des exemples de tonalités et les détails des procédures d'essai sont donnés dans les paragraphes 7 et 8/Q.724, dans le § 2.1.8/Q.764 et dans la Rec. UIT-T Q.1902.4

E.11 Paquetage de réseau

Identificateur de paquetage: nt (0x000b)

Version: 1

Extension: aucune

Ce paquetage définit les propriétés de terminaisons de réseau indépendantes du type de réseau.

E.11.1 Propriétés

Tampon maximal de gigue

Identificateur de propriété: jit (0x0007)

Cette propriété fixe une capacité maximale au tampon de gigue.

Type de paramètre: entier en millisecondes

Valeurs possibles: cette propriété est spécifiée en millisecondes.

Défini dans: LocalControl Descriptor

Caractéristiques: lecture/écriture

E.11.2 Evénements

Défaillance du réseau

Identificateur d'événement: netfail, 0x0005

La terminaison produit cet événement lorsqu'elle détecte une défaillance due à des raisons externes ou internes du réseau.

Paramètres de descripteur d'événement

Aucun

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor)

cause

Identificateur de paramètre: cs (0x0001)

Type de paramètre: String (chaîne)

Valeurs possibles: n'importe quelle chaîne texte

Ce paramètre peut être inclus avec l'événement de défaillance afin de fournir des informations de diagnostic sur la raison de la défaillance.

Alerte de qualité

Identificateur d'événement: qualert, 0x0006

Cette propriété permet à la passerelle MG d'indiquer une perte de qualité de la connexion réseau. A cette fin, la passerelle peut mesurer la perte de paquet, la gigue entre arrivées et le temps de propagation, puis indiquer ces valeurs sous la forme d'un pourcentage de perte de qualité.

Paramètres de descripteur d'événement

Seuil

Identificateur de paramètre: th (0x0001)

Type de paramètre: entier

Valeurs possibles: 0 à 99

Description: seuil relatif de la perte de qualité mesurée, calculé sur la base d'une méthode configurée qui peut prendre en compte, par exemple, la perte de paquets, la gigue et le temps de propagation. Cet événement est déclenché lorsque la valeur calculée dépasse le seuil.

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor)

Seuil

Identificateur de paramètre: th (0x0001)

Type de paramètre: entier

Valeurs possibles: 0 à 99

Description: seuil relatif de la perte de qualité mesurée, calculé sur la base d'une méthode configurée qui peut prendre en compte, par exemple, la perte de paquets, la gigue et le temps de propagation.

E.11.3 Signaux

Aucun.

E.11.4 Statistiques

Durée

Identificateur de statistiques: dur (0x0001)

Description: indique la durée pendant laquelle la terminaison a été dans le contexte.

Type de paramètre: nombre à double précision, en millisecondes

Octets envoyés

Identificateur de statistique: os (0x0002)

Type de paramètre: nombre à double précision

Valeurs possibles: tout entier de 64 bits

Octets reçus

Identificateur de statistique: or (0x0003)

Type de paramètre: nombre à double précision

Valeurs possibles: tout entier de 64 bits

E.11.5 Procédures

Aucune.

E.12 Paquetage du protocole de transport en temps réel (RTP)

Identificateur de paquetage: rtp (0x000c)

Version: 1

Extension: version 1 du paquetage de réseau

Ce paquetage est utilisé pour assurer le transfert en mode paquet de données multimédias au moyen du protocole de transport en temps réel (RTP, *real-time transport protocol*) (RFC 1889).

E.12.1 Propriétés

Aucune.

E.12.2 Evénements

Transition de charge utile

Identificateur d'événement: pltrans, 0x0001

Cet événement détecte et informe de la transition du format de charge utile RTP à un autre format.

Paramètres de descripteur d'événement

Aucun

Paramètres de descripteur d'événements observés (ObservedEventsDescriptor)

ParameterName: rtppayload

Identificateur de paramètre: rtppltype, 0x01

Type de paramètre: liste de types énumérés.

Valeurs possibles: la méthode de codage doit être spécifiée au moyen d'un ou de plusieurs noms de codage valides, comme définis dans le profil AV du protocole RTP ou comme enregistrés auprès de l'autorité IANA.

E.12.3 Signaux

Aucun.

E.12.4 Statistiques

Paquets envoyés

Identificateur de statistique: ps (0x0004)

Type de paramètre: nombre à double précision

Valeurs possibles: tout entier de 64 bits

Paquets reçus

Identificateur de statistique: pr (0x0005)

Type de paramètre: nombre à double précision

Valeurs possibles: tout entier de 64 bits

Perte de paquets

Identificateur de statistique: pl (0x0006)

Décrit le taux de perte de paquet actuel dans un flux RTP, comme défini dans le document RFC 1889. La perte de paquets est exprimée en pourcentage calculé comme suit: nombre de paquets perdus dans l'intervalle entre deux rapports de réception, divisé par le nombre de paquets attendus au cours de cet intervalle.

Type de paramètre: nombre à double précision

Valeurs possibles: nombre entier de 32 bits et fraction de 32 bits.

Gigue

Identificateur de statistique: jit (0x0007)

Demande la valeur actuelle de la gigue entre arrivées dans un flux RTP tel que défini par le document RFC 1889. La gigue mesure la variation du temps entre arrivées de paquets de données RTP.

Temps de propagation

Identificateur de statistique: delay (0x0008)

Demande la valeur actuelle du temps de propagation des paquets, exprimée en unités de marqueurs temporels. Même grandeur que la latence moyenne.

E.12.5 Procédures

Aucune.

E.13 Paquetage de circuit TDM

Identificateur de paquetage: tdmc (0x000d)

Version: 1

Extension: version 1 du paquetage de réseau

Ce paquetage peut être utilisé par toute terminaison qui prend en charge le gain et la limitation de l'écho. Il était initialement destiné à être utilisé sur les circuits TDM mais son utilisation peut être étendue.

Les nouvelles versions ou extensions de ce paquetage devraient pouvoir tenir compte des circuits non TDM.

E.13.1 Propriétés

Annulation d'écho

Identificateur de propriété: ec (0x0008)

Type de paramètre: opérateur booléen

Valeurs possibles:

"activée" (lorsque l'annulation d'écho est requise)

"désactivée" (lorsque l'annulation d'écho n'est pas requise.)

La valeur par défaut est "fournie".

Défini dans: LocalControlDescriptor

Caractéristiques: lecture/écriture

Commande de gain

Identificateur de propriété: gain (0x000a)

La commande de gain, ou l'utilisation de l'adaptation du niveau des signaux et de la réduction du niveau de bruit, est utilisée pour adapter le niveau du signal. Il est cependant nécessaire, par exemple pour les communications par modem, de désactiver cette fonction.

Type de paramètre: entier

Valeurs possibles:

le paramètre de commande de gain peut être, soit spécifié avec la valeur "automatique" (0xffffffff), soit sous forme de nombre explicite de décibels de gain (toute autre valeur d'entier). La valeur par défaut est fournie dans la passerelle MG.

Défini dans: LocalControlDescriptor

Caractéristiques: lecture/écriture

E.13.2 Evénements

Aucun.

E.13.3 Signaux

Aucun.

E.13.4 Statistiques

Aucune.

E.13.5 Procédures

Aucune.

Appendice I

Exemples de flux de communication

Tous les réalisateurs H.248.1 du protocole de la présente Recommandation doivent en lire attentivement la partie normative avant de l'appliquer à des implémentations. Aucun d'eux ne devrait utiliser les exemples donnés dans le présent appendice en tant qu'explications autonomes sur la façon de créer des messages de protocole.

Les exemples donnés dans le présent appendice font appel au protocole SDP pour le codage des descripteurs de flux Local et Remote. Le protocole SDP est défini dans le document RFC 2327. En cas de divergence dans les exemples entre le protocole SDP et le document RFC 2327, c'est ce dernier qui doit être consulté comme référence. Les profils audio utilisés sont définis dans le document RFC 1890 ou enregistrés auprès de l'autorité IANA. Par exemple, la loi A de la Rec. UIT-T G.711 est désignée par "MIC-A" dans le protocole SDP et est attribuée au profil 0. Le codage G.723.1 est nommé G723 et il correspond au profil 4; le codage H.263 est nommé H263 et il correspond au profil 34. Voir également http://www.iana.org/assignments/rtp-parameters.

I.1 Communication de passerelle résidentielle à passerelle résidentielle

Cet exemple de scénario décrit l'emploi des éléments du protocole afin d'établir une communication de passerelle résidentielle à passerelle résidentielle sur un réseau en protocole IP. Par concision, cet exemple part du principe que les deux passerelles résidentielles impliquées dans la communication sont régies par le même contrôleur MGC.

I.1.1 Programmation du comportement de repos de terminaisons de ligne analogique passant par une passerelle résidentielle

On trouvera ci-dessous la description des invocations d'interface API issues du contrôleur MGC et des passerelles MG pour faire en sorte que les terminaisons de ce scénario soient programmées au comportement de repos. La passerelle MG d'origine comme son homologue de destination a des terminaisons de ligne analogique au repos qui sont programmées pour rechercher les événements de lancement d'appel (c'est-à-dire de décrochage) au moyen de la commande Modify contenant les paramètres appropriés. Le contexte néant est utilisé pour indiquer que les terminaisons ne sont pas encore associées à un contexte. La terminaison racine est utilisée pour renvoyer à la passerelle MG entière et non pas à une terminaison contenue dans cette passerelle.

Dans cet exemple, la passerelle MG1 possède l'adresse IP 124.124.124.222. La passerelle MG2 possède l'adresse IP 125.125.125.111 et le contrôleur MGC 123.123.123.4. L'accès Megaco par défaut est 55555 pour les trois entités.

1) Une passerelle MG s'enregistre avec un contrôleur MGC au moyen de la commande ServiceChange:

```
MG1 à MGC:

MEGACO/1 [124.124.124.222]
Transaction = 9998 {
    Context = - {
        ServiceChange = ROOT {Services {
            Method=Restart,
            ServiceChangeAddress=55555, Profile=ResGW/1}
        }
    }
}
```

2) Le contrôleur MGC envoie la réponse suivante:

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Reply = 9998 {
   Context = - {ServiceChange = ROOT {
     Services {ServiceChangeAddress=55555, Profile=ResGW/1} } }
}
```

3) Le contrôleur MGC programme une terminaison dans le contexte NULL. L'identificateur de cette terminaison est A4444, l'identificateur de flux est 1, l'identificateur de demande contenu dans le descripteur d'événements est 2222. L'identificateur mId est celui de l'expéditeur de ce message, dans ce cas, il s'agit de l'adresse IP et de l'accès [123.123.123.4]:55555. Le paramètre Mode pour ce flux est réglé à SendReceive (envoyer et recevoir). Le paramètre "al" est le paquetage de supervision de la ligne analogique. On suppose que tant l'option Local que Remote sont assurées.

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 9999 {
    Context = - {
        Modify = A4444 {
            Media { Stream = 1 {
                     LocalControl {
                         Mode = SendReceive,
                         tdmc/gain=2,  ; in dB,
                         tdmc/ec=on
                     },
                 }
            },
            Events = 2222 {al/of (strict=state)}
        }
    }
}
```

Le script de numérotation peut avoir été chargé préalablement dans la passerelle MG. Sa fonction sera d'attendre le décrochage, d'activer la tonalité d'invitation et de commencer à acquérir des chiffres par tonalités DTMF. Dans cet exemple cependant, l'on utilise le script de numérotation, qui est mis en place après la détection du décrochage (étape 5) ci-dessous).

Noter que le descripteur d'événements incorporé peut avoir été utilisé pour combiner les étapes 3) et 4) avec les étapes 8) et 9), ce qui élimine les étapes 6) et 7).

4) La passerelle MG1 accepte la commande Modify avec cette réponse:

```
MG1 à MGC:
MEGACO/1 [124.124.124.222]:55555
Reply = 9999 {
   Context = - {Modify = A4444}
}
```

5) Un échange similaire se déroule entre la passerelle MG2 et le contrôleur MGC, ce qui produit une terminaison au repos appelée A5555.

I.1.2 Acquisition des chiffres de l'expéditeur et de la terminaison d'origine

La procédure suivante est fondée sur les conditions indiquées ci-dessus. Elle décrit les transactions issues du contrôleur MGC et de la passerelle média d'origine (MG1, *originating media gateway*) afin d'acquérir la terminaison d'origine (A4444) en passant par les étapes d'acquisition de chiffres nécessaires pour ouvrir une connexion vers la passerelle média de destination (MG2, *terminating media gateway*).

6) La passerelle MG1 détecte un événement de décrochage en provenance de l'utilisateur 1 et le signale au contrôleur MGC au moyen de la commande Notify.

```
MG1 à MGC:
MEGACO/1 [124.124.124.222]:55555
Transaction = 10000 {
   Context = - {
       Notify = A4444 {ObservedEvents =2222 {
         19990729T22000000:al/of(init=false)}}
   }
}
```

7) Et la commande Notify est acquittée.

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Reply = 10000 {
    Context = - {Notify = A4444}
}
```

8) Le contrôleur MGC modifie la terminaison afin de jouer la tonalité de numérotation, de rechercher des chiffres conformément à Dialplan0 et de chercher maintenant un événement de raccrochage.

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10001 {
    Context = - {
        Modify = A4444 {
            Events = 2223 {
                al/on(strict=state), dd/ce {DigitMap=Dialplan0}
            },
            Signals {cg/dt},
            DigitMap= Dialplan0{
(0| 00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxxx|9011x.)}
        }
    }
}
```

9) Et la commande Modify est acquittée.

```
MG1 à MGC:
MEGACO/1 [124.124.124.222]:55555
Reply = 10001 {
    Context = - {Modify = A4444}
}
```

10) Ensuite, les chiffres sont accumulés par la passerelle MG1 au fur et à mesure qu'ils sont composés par l'utilisateur 1. La tonalité de numération (Dialtone) est arrêtée à la détection du premier chiffre. Lorsqu'une correspondance appropriée est réalisée pour les chiffres recueillis avec le plan de numérotation couramment programmé (Dialplan) pour A4444, une autre commande Notify est envoyée au contrôleur de passerelle média.

```
MG1 à MGC:
MEGACO/1 [124.124.124.222]:55555
Transaction = 10002 {
   Context = - {
       Notify = A4444 {ObservedEvents =2223 {
         19990729T22010001:dd/ce{ds="916135551212",Meth=UM}}}
   }
}
```

11) Et la commande Notify est acquittée.

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Reply = 10002 {
    Context = - {Notify = A4444}
}
```

12) Le contrôleur analyse ensuite les chiffres et détermine qu'une connexion doit être établie de la passerelle MG1 à la passerelle MG2. Aussi bien la terminaison TDM A4444 qu'une terminaison RTP sont ajoutées à un nouveau contexte dans la passerelle MG1. Le mode est réception seulement car les valeurs du descripteur Remote ne sont pas encore spécifiées. Les codecs préférés sont énumérés dans l'ordre de préférence du contrôleur MGC.

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10003 {
    Context = $ {
       Add = A4444,
       Add = $ {
           Media {
             Stream = 1 {
                  LocalControl {
                      Mode = ReceiveOnly,

                      nt/jit=40 ; in ms
                  },
                  Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
                  }
             }
          }
       }
    }
}
```

NOTE – Le contrôleur MGC indique ses valeurs de paramètre préférées comme une série de blocs SDP contenue dans le descripteur Local. Celle-ci insère le descripteur Local dans la réponse**.**

13) La passerelle MG1 acquitte la nouvelle terminaison et insère l'adresse IP locale et l'accès UDP. Elle effectue également, dans le descripteur Local, un choix de codec sur la base des préférences du contrôleur MGC. La passerelle MG1 règle l'accès RTP à la valeur 2222.

```
MEGACO/1 [124.124.124.222]:55555
Reply = 10003 {
   Context = 2000 {
      Add = A4444,
      Add=A4445{
         Media {
             Stream = 1 {
                 Local {
v=0
o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
```

```
a=recvonly
                    } ; le profil RTP pour G.723.1 est 4
                }
            }
        }
    }
}
```

14) Le contrôleur MGC associera désormais la terminaison A5555 avec un nouveau contexte dans la passerelle MG2 et établira vers l'utilisateur d'origine (utilisateur 1) une connexion d'émission-réception avec identification dans le flux RTP (c'est-à-dire que la terminaison A5556 sera assignée). Le contrôleur MGC règle également la sonnerie sur A5555.

```
MGC to MG2:
MEGACO/1 [123.123.123.4]:55555
Transaction = 50003 {
    Context = $ {
       Add = A5555  { Media {
            Stream = 1 {
                 LocalControl {Mode = SendReceive} }},
       Events=1234{al/of(strict=state)},
            Signals {al/ri}
            },
        Add  = $ {Media {
            Stream = 1 {
                 LocalControl {
                    Mode = SendReceive,
                    nt/jit=40 ; in ms
                 },
                 Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
                 },
                 Remote {
v=0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
                    } ; le profil RTP pour G.723.1 est 4
                }
            }
       }
    }
}
```

15) Cette connexion est acquittée. Le numéro d'accès au flux est différent du numéro d'accès de commande. Dans ce cas, ce numéro est 1111 (dans le protocole SDP).

```
MG2 à MGC:
MEGACO/1 [125.125.125.111]:55555
Reply = 50003 {
   Context = 5000 {
     Add = A5555,
      Add = A5556{
         Media {
            Stream = 1 {
                Local {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
```

```
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
}
            } ; le profil RTP pour G.723.1 est 4
          }
        }
   }
}
```

16) L'adresse IP et l'accès UDP ci-dessus définis doivent maintenant être indiqués à la passerelle MG1.

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10005 {
  Context = 2000 {
    Modify = A4444 {
      Signals {cg/rt}
    },
    Modify = A4445 {
       Media {
            Stream = 1 {
                Remote {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
                }
            } ; le profil RTP pour G.723.1 est 4
        }
    }
  }
}

MG1 à MGC:
MEGACO/1 [124.124.124.222]:55555
Reply = 10005 {
   Context = 2000 {Modify = A4444, Modify = A4445}
}
```

17) Les deux passerelles sont maintenant connectées et l'utilisateur 1 entend le retour d'appel. La passerelle MG2 attend maintenant que l'utilisateur 2 soulève le combiné pour que la communication soit établie dans les deux sens.

```
De MG2 à MGC:

MEGACO/1 [125.125.125.111]:55555
Transaction = 50005 {
   Context = 5000 {
       Notify = A5555 {ObservedEvents =1234 {
         19990729T22020002:al/of(init=false)}}
   }
}

De MGC à MG2:

MEGACO/1 [123.123.123.4]:55555
Reply = 50005 {
    Context = - {Notify = A5555}
}
```

```
De MGC à MG2:

MEGACO/1 [123.123.123.4]:55555
Transaction = 50006 {
   Context = 5000 {
      Modify = A5555 {
         Events = 1235 {al/on(strict=state)},
          Signals { } ; pour couper la sonnerie
      }
   }
}

De MG2 à MGC:

MEGACO/1 [125.125.125.111]:55555
Reply = 50006 {
 Context = 5000 {Modify = A4445}
}
```

18) Dans la passerelle MG1, le mode est commuté sur émission-réception et le retour d'appel est arrêté.

```
MGC à MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10006 {
   Context = 2000 {
      Modify = A4445 {
         Media {
            Stream = 1 {
               LocalControl {
                  Mode=SendReceive
               }
            }
         }
      },
      Modify = A4444 {
         Signals { }
      }
   }
}

De MG1 à MGC:
MEGACO/1 [124.124.124.222]:55555
Reply = 10006 {
   Context = 2000 {Modify = A4445, Modify = A4444}}
```

19) Le contrôleur MGC décide d'envoyer la commande Audit à la terminaison RTP de la passerelle MG2.

```
MEGACO/1 [123.123.123.4]:55555
Transaction = 50007 {
   Context = - {AuditValue = A5556{
      Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
   }
}
```

20) La passerelle MG2 répond.

```
MEGACO/1 [125.125.125.111]:55555
Reply = 50007 {
   Context = - {
AuditValue = A5556 {
          Media {
```

```
            TerminationState { ServiceStates = InService,
                Buffer = OFF },
            Stream = 1 {
                LocalControl { Mode = SendReceive,
                   nt/jit=40 },
                Local {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP  4
a=ptime:30
               },
                Remote {
v=0

o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP  4
a=ptime:30
                } } },
          Events,
          Signals,
          DigitMap,
          Packages {nt-1, rtp-1},
          Statistics { rtp/ps=1200,  ; paquets envoyés
                       nt/os=62300, ; octets envoyés
                       rtp/pr=700, ; paquets reçus
                       nt/or=45100, ; octets reçus
                       rtp/pl=0.2,  ; % perte de paquets
                       rtp/jit=20,
                       rtp/delay=40 } ; latence moyenne
      }
    }
}
```

21) Lorsque le contrôleur MGC reçoit un signal de raccrochage en provenance d'une des passerelles MG, il libère la communication. Dans cet exemple, l'utilisateur situé du côté de la passerelle MG2 raccroche le premier.

```
De MG2 à MGC:

MEGACO/1 [125.125.125.111]:55555
Transaction = 50008 {
   Context = 5000 {
       Notify = A5555 {ObservedEvents =1235 {
          19990729T24020002:al/on(init=false)}
       }
   }
}

De MGC à MG2:

MEGACO/1 [123.123.123.4]:55555
Reply = 50008 {
    Context = - {Notify = A5555}
}
```

22) Le contrôleur MGC envoie ensuite aux deux passerelles MG une commande Subtract afin de libérer la communication. Seules les commandes de soustraction envoyées à la passerelle MG2 sont représentées ici. Chaque terminaison possède son propre paquetage de statistiques recueillies. Un contrôleur MGC peut ne pas avoir besoin de demander que les deux commandes lui soient renvoyées. La terminaison A5555 est physique, tandis que la terminaison A5556 est en protocole RTP.

```
De MGC à MG2:

MEGACO/1 [123.123.123.4]:55555
Transaction = 50009 {
   Context = 5000 {
      Subtract = A5555 {Audit{Statistics}},
      Subtract = A5556 {Audit{Statistics}}
   }
}

De MG2 à MGC:

MEGACO/1 [125.125.125.111]:55555
Reply = 50009 {
   Context = 5000 {
     Subtract = A5555 {
          Statistics {
             nt/os=45123, ; Octets envoyés
             nt/dur=40 ; en secondes
             }
       },
       Subtract = A5556 {
          Statistics {
             rtp/ps=1245, ; paquets envoyés
             nt/os=62345, ; octets envoyés
             rtp/pr=780, ; paquets reçus
             nt/or=45123, ; octets reçus
             rtp/pl=10, ;  % perte de paquets
             rtp/jit=27,
             rtp/delay=48 ; latence moyenne
          }
       }
   }
}
```

23) Le contrôleur MGC configure ensuite les deux passerelles MG1 et MG2 de façon qu'elles soient prêtes à détecter le prochain événement de décrochage. Se reporter à l'étape 1). Noter que cet état pourrait être défini comme étant la valeur par défaut d'une terminaison dans le contexte néant. Dans ce cas, aucun message n'a besoin d'être envoyé du contrôleur MGC à la passerelle MG. Une fois qu'une terminaison revient au contexte néant, elle reprend ses valeurs par défaut assignées.

SÉRIES DES RECOMMANDATIONS UIT-T

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

23025

Imprimé en Suisse
Genève, 2003