



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Q.823

(07/96)

SÉRIE Q: COMMUTATION ET SIGNALISATION

Spécifications du système de signalisation n° 7 –
Interface Q3

**Spécifications fonctionnelles d'étape 2 et
d'étape 3 de la gestion de trafic**

Recommandation UIT-T Q.823

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE Q

COMMUTATION ET SIGNALISATION

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DES SYSTÈMES DE SIGNALISATION N° 4 ET N° 5	Q.120–Q.249
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 6	Q.250–Q.309
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R1	Q.310–Q.399
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R2	Q.400–Q.499
COMMUTATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.849
Généralités	Q.700
Sous-système transport de messages	Q.701–Q.709
Sous-système commande des connexions sémaphores	Q.711–Q.719
Sous-système utilisateur téléphonie	Q.720–Q.729
Services complémentaires du RNIS	Q.730–Q.739
Sous-système utilisateur données	Q.740–Q.749
Gestion du système de signalisation n° 7	Q.750–Q.759
Sous-système utilisateur du RNIS	Q.760–Q.769
Sous-système application de gestion des transactions	Q.770–Q.779
Spécifications d'essais	Q.780–Q.799
Interface Q3	Q.800–Q.849
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
Généralités	Q.850–Q.919
Couche liaison de données	Q.920–Q.929
Couche réseau	Q.930–Q.939
Gestion usager-réseau	Q.940–Q.949
Description de l'étape 3 pour les services complémentaires utilisant le système DSS 1	Q.950–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1999
RNIS À LARGE BANDE	Q.2000–Q.2999

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

RECOMMANDATION UIT-T Q.823

SPECIFICATIONS FONCTIONNELLES D'ETAPE 2 ET D'ETAPE 3 DE LA GESTION DE TRAFIC

Résumé

La présente Recommandation vise à fournir un modèle d'information qui englobe les aspects de gestion des fonctions du service de gestion de trafic dans un commutateur. Son domaine d'application se limite aux réseaux commutés de circuits qui utilisent l'acheminement hiérarchique.

La présente Recommandation est centrée sur la description d'étape 2 et d'étape 3 de l'interface Q3 entre les éléments de réseaux (NE, *network elements*) et les systèmes d'exploitation (OS, *operations systems*). La description d'étape 1 figure dans les Recommandations de la série E.410 et dans la Recommandation E.502.

Source

La Recommandation UIT-T Q.823, élaborée par la Commission d'études 11 (1993-1996) de l'UIT-T, a été approuvée le 9 juillet 1996 selon la procédure définie dans la Résolution n° 1 de la CMNT.

AVANT-PROPOS

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

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

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

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en oeuvre 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 avait/n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en oeuvre 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 oeuvre de consulter la base de données des brevets du TSB.

© UIT 1997

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

TABLE DES MATIÈRES

	Page
1	Domaine d'application..... 1
2	Introduction 1
2.1	Prescriptions fonctionnelles 2
2.2	Méthode de modélisation 2
3	Références normatives 2
4	Définitions..... 3
4.1	Acheminement en débordement..... 3
4.2	Acheminement sur route de premier choix 4
5	Symboles et abréviations..... 4
6	Modèle d'information 5
6.1	Diagramme entité-relation..... 5
6.2	Arborescence d'héritage..... 7
6.3	Arborescence de dénomination 8
7	Description du modèle d'information 10
7.1	Description de la classe d'objets..... 10
7.1.1	Fragment élément géré..... 10
7.1.2	Fragment surveillance des performances 12
7.1.3	Fragment commande..... 23
7.1.4	Fragment état..... 42
7.1.5	Fragment Administratif..... 44
7.2	Définition des attributs communs 44
7.2.1	Nom distinctif relatif..... 44
7.2.2	Attributs d'état 44
7.3	Description des actions..... 44
7.4	Descriptions des notifications 45
8	Unités fonctionnelles..... 45
8.1	Unités fonctionnelles définies dans d'autres Recommandations..... 46
8.2	Négociation des unités fonctionnelles..... 46
9	Relations avec les autres Recommandations..... 46
10	Conformité 47
10.1	Conformité statique 47
10.2	Conformité dynamique..... 48

10.3	Prescription en matière de déclaration de conformité de l'implémentation de gestion.....	48
11	Définitions formelles des classes d'objets	49
11.1	Définition des classes d'objets.....	49
11.1.1	ACC	49
11.1.2	accAffectedTraffic	50
11.1.3	accTrigger.....	50
11.1.4	adc	50
11.1.5	adcTrigger	52
11.1.6	cancelFrom.....	53
11.1.7	cancelRerouted.....	53
11.1.8	cancelTo	54
11.1.9	CircuitEndPointSubgroupCurrentData	54
11.1.10	circuitEndPointSubgroupHistoryData.....	55
11.1.11	congestionLevelIndication	55
11.1.12	destinationCodeControl	56
11.1.13	dccGroup.....	57
11.1.14	exchangeCurrentData.....	58
11.1.15	exchangeHistoryData	58
11.1.16	htrDestination.....	59
11.1.17	observedDestination.....	59
11.1.18	observedDestinationCurrentData	60
11.1.19	observedDestinationHistoryData	60
11.1.20	scr	61
11.1.21	scrAffectedTraffic	62
11.1.22	skip	62
11.1.23	StateIndicator	63
11.1.24	tarFrom.....	64
11.1.25	tarTo	65
11.1.26	tmCircuitEndPointSubgroup.....	66
11.1.27	tmCircuitEndPointSubgroupCurrentData	66
11.1.28	tmExchangeCurrentData	66
11.1.29	tmObservedDestinationCurrentData.....	67
11.1.30	tmTrafficControlCurrentData	67
11.1.31	trafficControl.....	68
11.1.32	trafficControlCurrentData	68
11.1.33	trafficControlHistoryData	69
11.2	Définition de corrélation de nom.....	69
11.3	Définition des progiciels	73
11.4	Définition du comportement commun	77

	Page
11.5	Definition of attributes 77
11.6	Définition des Actions..... 87
11.7	Définition des avis..... 88
11.8	Module des types définis ASN.1 89
Appendice I – Service de gestion des réseaux de gestion des télécommunications: gestion du trafic..... 94	
Appendice II – Corrélation entre les commandes trafic définies dans la Recommandation E.412 et les classes d'objets définies dans la présente Recommandation..... 96	
Appendice III – Diagrammes de transition d'état des commandes automatiques et manuelles..... 97	
Appendice IV – Emploi des commandes d'annulation du débordement vers (<i>CancelTo</i>) et d'annulation du débordement depuis (<i>CancelFrom</i>) aux trafics de premier choix et de débordement 98	
Appendice V – Utilisation de la classe de données chronologiques, <i>historyData</i> , définie dans la Recommandation Q.822 pour les mesures du trafic 99	
V.1	Introduction 99
V.2	Aperçu général du modèle d'information de gestion des performances..... 99
V.3	Configuration initiale 100
	V.3.1 Données courantes 100
	V.3.2 Données chronologiques 100
	V.3.3 Releveur simple..... 100
V.4	Rapport émis de façon autonome 101
V.5	Rapport à la demande 102
V.6	Changement du rapport 102
V.7	Conclusions 102
Appendice VI – Définition de l'objet <i>CircuitEndPointSubgroup</i> (sous-faisceau d'extrémité de circuits) 104	
Appendice VII – Mappage des fonctions de gestion du réseau sur les unités fonctionnelles. 106	
Appendice VIII – Prise en charge de la fonctionnalité de vérification avec le modèle décrit dans la Recommandation Q.823..... 108	
VIII.1	Introduction 108
VIII.2	Prescriptions fonctionnelles 108
VIII.3	Types de vérification..... 108
	VIII.3.1 Vérification complète ou partielle..... 108
	VIII.3.2 Vérification des modifications 109
VIII.4	Mécanismes 109

	Page
VIII.4.1 Utilisation du service PT-GET	109
VIII.4.2 Utilisation du releveur simple	110
VIII.4.3 Utilisation de l'objet de découverte <i>discoveryObject</i>	111
VIII.5 Conclusions	111
Appendice IX – Exemples d'erreurs d'adressage dans la Recommandation Q.823.....	112
IX.1 Introduction	112
IX.2 Erreurs générées par la machine protocolaire.....	112
IX.3 Utilisation d'erreur CMISE.....	112
IX.4 Utilisation des erreurs spécifiques.....	114
IX.5 Conclusion.....	115
Appendice X – Hiérarchie des commandes.....	115

Recommandation UIT-T Q.823

SPECIFICATIONS FONCTIONNELLES D'ETAPE 2 ET D'ETAPE 3 DE LA GESTION DE TRAFIC

(Genève, 1996)

1 Domaine d'application

La présente Recommandation vise à fournir un modèle d'information qui englobe les aspects de gestion des fonctions du service de gestion de trafic dans un commutateur. Son domaine d'application se limite aux réseaux commutés de circuits qui utilisent l'acheminement hiérarchique.

La présente Recommandation est centrée sur la description d'étape 2 et d'étape 3 de l'interface Q3 entre les éléments de réseaux et les systèmes d'exploitation. La description d'étape 1 figure dans les Recommandations de la série E.410 et dans la Recommandation E.502.

On doit appliquer les restrictions suivantes au domaine d'application:

- pour la surveillance et la gestion de trafic, la présente Recommandation utilise les mesures de trafic qui sont définies dans les Recommandations de la série E;
- le traitement des informations de trafic au niveau du système d'exploitation (OS) ainsi que l'envoi de ces données au niveau de ce système dépassent le cadre de la présente Recommandation;
- les données de performance du réseau sont nécessaires en entrée des commandes de gestion du trafic. Quelquefois il est possible de les utiliser pour les mesures de trafic mais on ignore cette coïncidence dans ce contexte. Ces données sont identifiées et modélisées dans la mesure où cela s'impose et lorsque les modèles ne sont pas disponibles dans d'autres Recommandations;
- la présente Recommandation ne redéfinit pas les fonctions que les modèles d'information définis dans d'autres Recommandations peuvent prendre en charge. Ces modèles sont référencés directement par insertion d'objets ou indirectement par définition de sous-classes à partir d'autres informations;
- dans la mesure du possible, on utilise les fonctions de gestion-systèmes OSI.

2 Introduction

Le but de la gestion de trafic est d'établir sans difficulté le plus possible d'appels. Cet objectif est atteint en maximisant l'utilisation des ressources disponibles d'une situation. Il appartient aussi à cette fonction de superviser les performances d'un réseau et d'être capable, si nécessaire, d'intervenir pour commander le flux de trafic afin d'optimiser l'utilisation des capacités du réseau.

Le modèle d'information présenté dans la présente Recommandation offre une approche commune d'extraction des données de performance à partir de l'élément de réseau, et de gestion des commandes et des instructions, que le système d'exploitation envoie à cet élément. Les données de performance du réseau fournissent une information d'activation des commandes d'administration du trafic et de validation des actions prises en ce qui concerne la gestion de trafic. Elles servent aussi en tant que données d'entrée pour les autres actions de gestion de trafic.

2.1 Prescriptions fonctionnelles

Les fonctions de commande de gestion du trafic considérées dans ce modèle dépendent des commandes spécifiées dans la Recommandation E.412.

Les fonctions de surveillance des performances de ce modèle dépendent des principes de mesure traités dans la Recommandation E.502.

L'Appendice I identifie les fonctions de gestion de trafic qui sont définies dans la Recommandation M.3200 et qui sont traitées par la présente Recommandation ainsi que celles qui dépassent le cadre de la présente Recommandation.

L'Appendice II fournit un mappage des commandes décrites dans la Recommandation E.412 sur les objets de commande traités dans la présente Recommandation.

2.2 Méthode de modélisation

On utilise dans la présente Recommandation les techniques de modélisation décrites dans la Recommandation M.3020 (Méthodologie pour la spécification des interfaces du réseau de gestion des télécommunications).

Pour la modélisation, on a adopté les définitions des classes d'objets gérés utilisées qui sont données dans les Recommandations OSI (série X.700).

3 Références normatives

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 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.

- Recommandation E.410 du CCITT (1992), *Gestion du réseau international – Informations générales.*
- Recommandation E.411 du CCITT (1992), *Gestion du réseau international – Directives d'exploitation.*
- Recommandation E.412 du CCITT (1992), *Commandes de gestion du réseau.*
- Recommandation E.502 du CCITT (1992), *Spécifications des mesures de trafic relatives aux commutateurs de télécommunications numériques.*
- Recommandation UIT-T M.3010 (1996), *Principes des réseaux de gestion des télécommunications.*
- Recommandation UIT-T M.3100 (1995), *Modèle générique d'information de réseau.*
- Recommandation UIT-T Q.542 (1993), *Objectifs nominaux des commutateurs numériques – Exploitation et maintenance.*
- Recommandation UIT-T Q.763 (1993), *Formats et codes du sous-système utilisateur pour le RNIS du système de signalisation n° 7.*
- Recommandation UIT-T Q.822 (1994), *Description d'étape 1, d'étape 2 et d'étape 3 de l'interface Q3 – Gestion de la qualité de fonctionnement.*

- Recommandation X.720 du CCITT (1992), *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: modèle d'information de gestion.*
- Recommandation X.721 du CCITT (1992), *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: définition des informations de gestion.*
- Recommandation X.722 du CCITT (1992), *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: directives pour la définition des objets gérés.*
- Recommandation X.730 du CCITT (1992), *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion des objets.*
- Recommandation X.731 du CCITT (1992), *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion d'états.*
- Recommandation X.734 du CCITT (1992), *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion des rapports d'événement.*
- Recommandation UIT-T X.738 (1993), *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de récapitulation.*

4 Définitions

La présente Recommandation définit les termes suivants.

4.1 Acheminement en débordement

4.1.1 trafic de débordement: ensemble d'appels offerts à un sous-faisceau de circuits qui n'est pas de premier choix; cette procédure s'appelle le débordement.

4.1.2 signal de réponse: information émise vers l'arrière pour indiquer qu'il a été répondu à l'appel. Ce terme est défini dans l'Annexe A/E.410.

4.1.3 tentative: tentative pour obtenir un circuit sur un sous-faisceau de circuits ou vers une destination. Une tentative peut aboutir ou ne pas aboutir à la prise d'un circuit sur ledit faisceau de circuits ou ladite destination. Ce terme est défini dans l'Annexe A/E.410.

4.1.4 circuit: moyen de transmission qui permet une communication entre deux commutateurs. Ce terme est défini dans l'Annexe A/E.410.

4.1.5 faisceau de circuits: ensemble des circuits commutés qui relient directement entre eux deux centres de commutation. Ce terme est défini dans l'Annexe A/E.410. Pour la gestion du trafic, un faisceau de circuit est invisible; c'est-à-dire qu'il n'est pas modélisé au niveau de la gestion des éléments de réseaux.

4.1.6 sous-faisceau de circuits: ensemble de circuits d'un faisceau de circuits que l'on peut identifier pour des raisons d'ordre technique ou d'exploitation. Un faisceau de circuits peut comprendre un ou plusieurs sous-faisceaux de circuits. Ce terme est défini dans l'Annexe A/E.410.

4.1.7 destination: pays, zone, commutateur ou autre emplacement, ou encore service spécial dans lequel l'abonné demandé est situé et qui peut être spécifié dans le pays. Une destination est identifiée par un ou plusieurs indicatifs de destination. Ce terme est défini dans l'Annexe A/E.410.

4.1.8 indicatif de destination: un indicatif de destination est identifié par les chiffres utilisés pour analyser et traiter la communication. Il peut être défini avec la précision nécessaire. Ce terme est défini dans l'Annexe A/E.410.

4.2 Acheminement sur route de premier choix

4.2.1 trafic de premier choix: ensemble d'appels offerts à un sous-faisceau de premier choix.

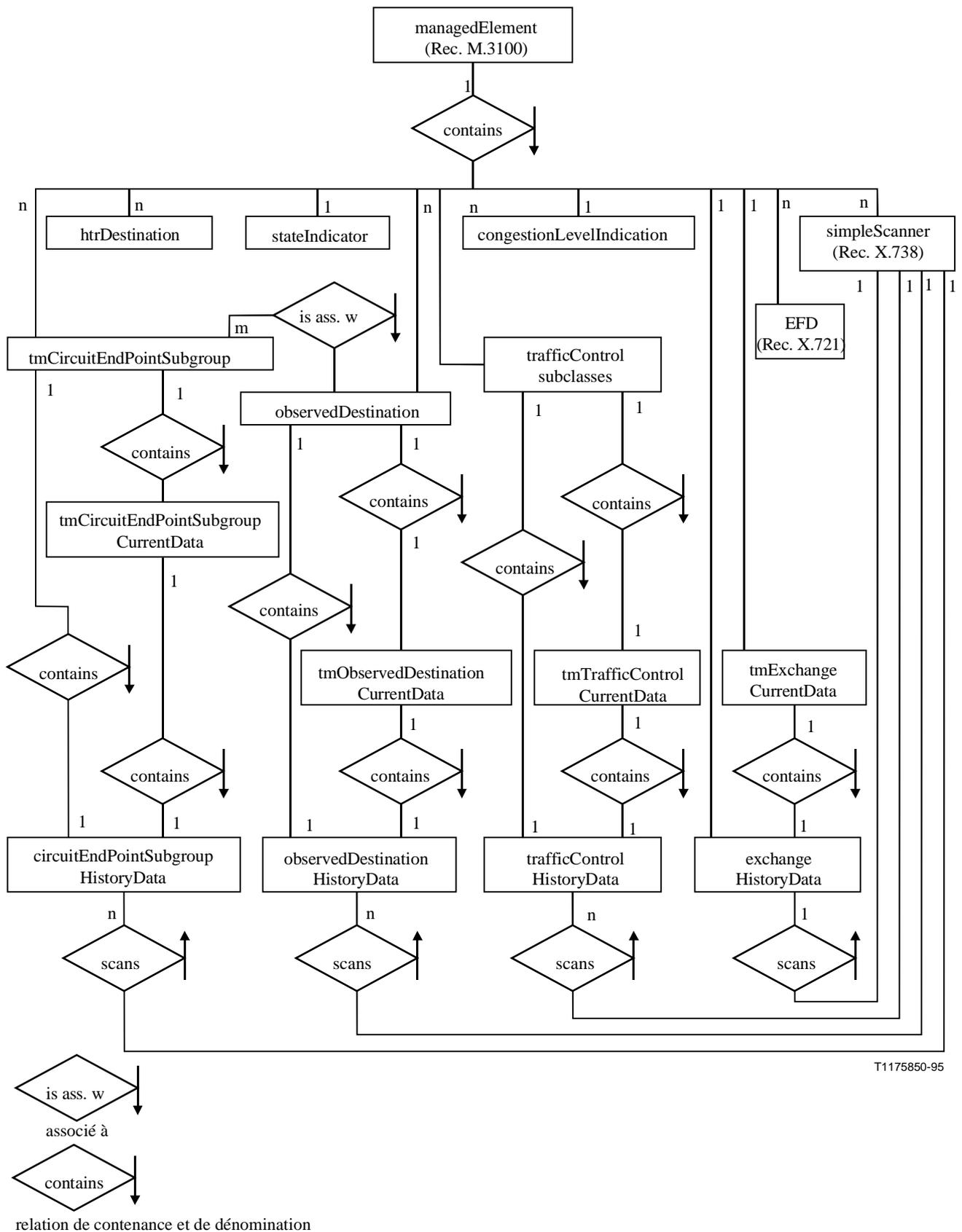
4.2.2 prise: tentative de prise d'un circuit sur un faisceau de circuits, qui aboutit à l'obtention d'un circuit sur ce faisceau de circuits. Ce terme est défini dans l'Annexe A/E.410.

4.2.3 nœud de commutation: commutateur représenté par une instance d'élément géré ou par une sous-classe d'élément géré.

5 Symboles et abréviations

La liste suivante regroupe les symboles et abréviations utilisés dans la présente Recommandation; elle ne comporte pas les noms des classes d'objets gérés et des attributs qui figurent dans les définitions formelles de la présente Recommandation.

ABR	taux de tentatives de prise avec réponse (<i>answer bid ratio</i>)
ACC	réduction automatique de l'encombrement (<i>automatic congestion control</i>)
ADC	commande automatique de destination (<i>automatic destination control</i>)
ASR	taux de prises avec réponse (<i>answer seizure ratio</i>)
HTR	difficile à atteindre: difficile d'accès (<i>hard-to-reach</i>)
Id	identificateur
MIB	base d'informations de gestion (<i>management information base</i>)
NE	élément de réseau (<i>network element</i>)
OA&M	gestion, exploitation et maintenance (<i>operation, administration and maintenance</i>)
OS	système d'exploitation (<i>operations system</i>)
OSI	interconnexion des systèmes ouverts (<i>open systems interconnection</i>)
RDN	nom distinctif relatif (<i>relative distinguished name</i>)
RNIS	réseau numérique à intégration de services
RTPC	réseau téléphonique public commuté

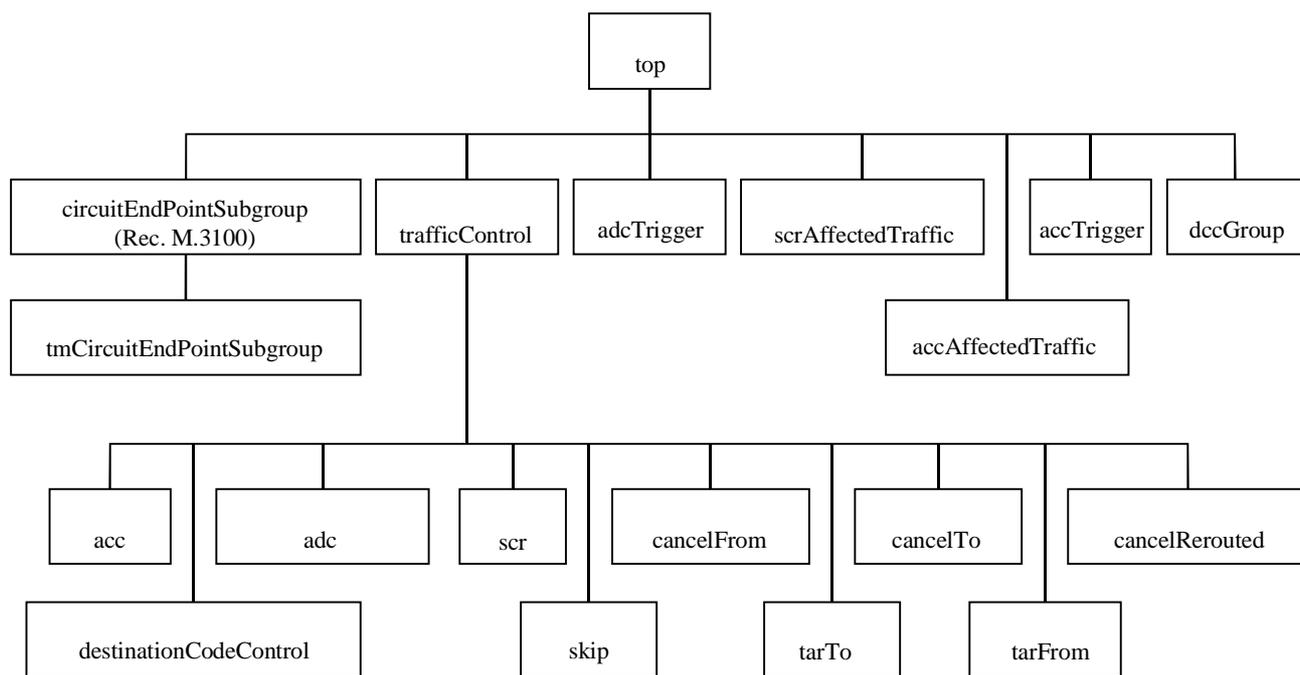


T1175850-95

Figure 6-2/Q.823 – Diagramme entité-relation des fragments élément géré, performance, état et administration

6.2 Arborescence d'héritage

Voir les Figures 6-3 et 6-4.



T1175860-95

Figure 6-3/Q.823 – Arborescence d'héritage du fragment commande

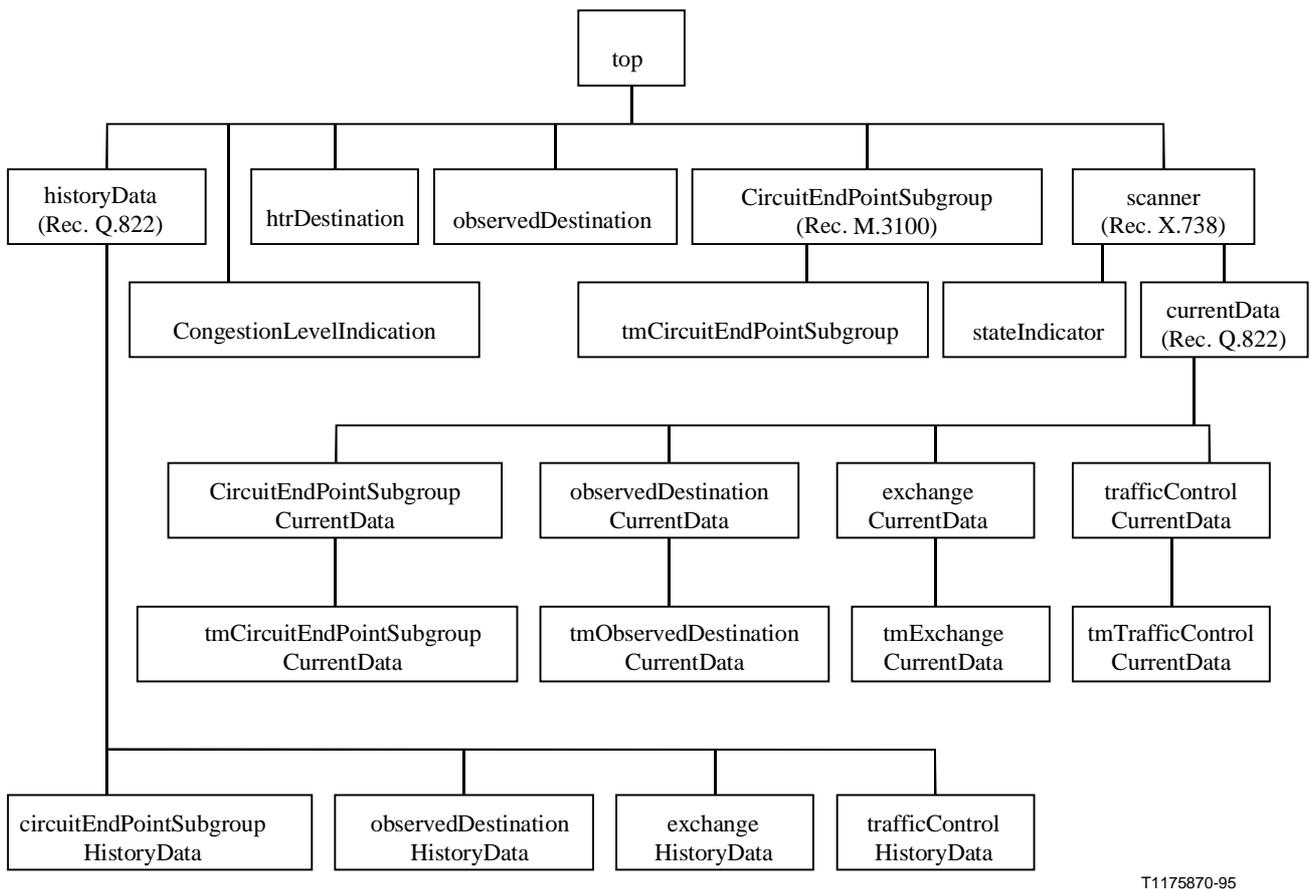


Figure 6-4/Q.823 – Arborecence d'héritage des fragments élément géré, performance, état et administration

6.3 Arborecence de dénomination

Voir les Figures 6-5 et 6-6.

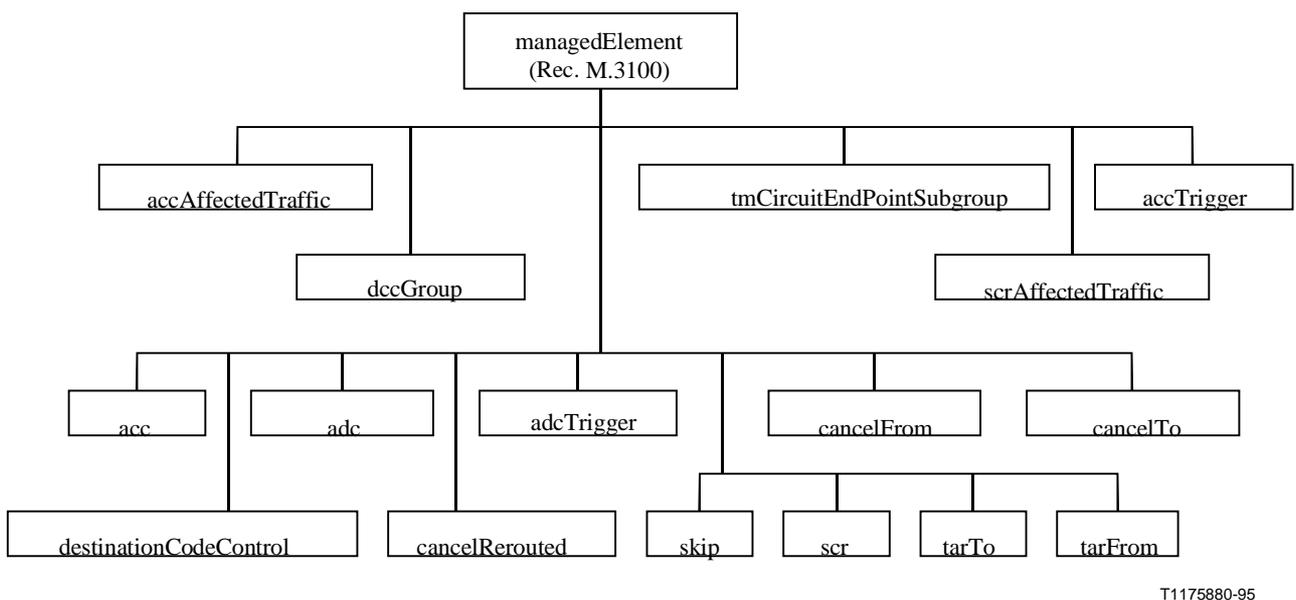
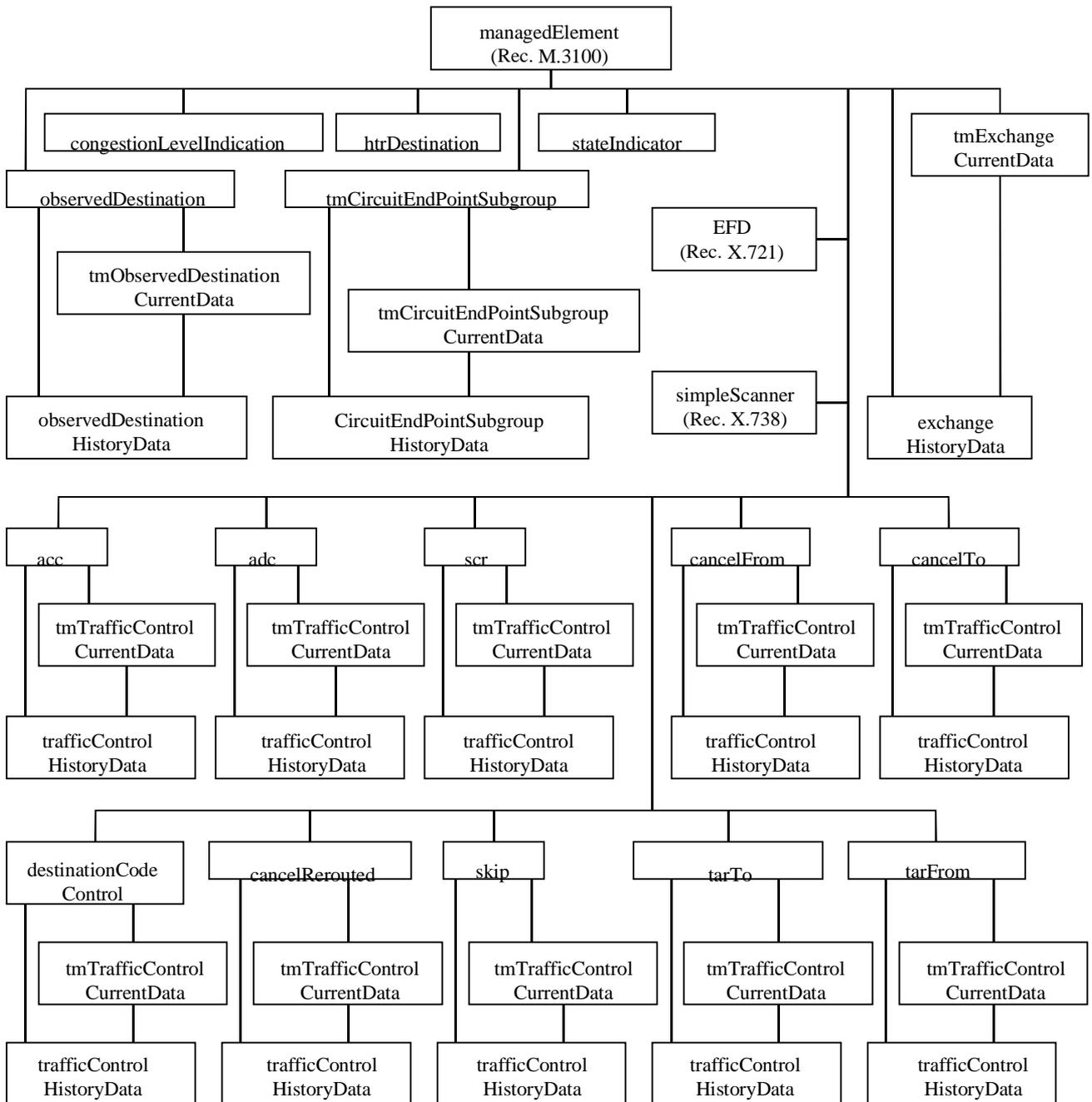


Figure 6-5/Q.823 – Corrélation de nom pour le fragment commande



T1175890-95

Figure 6-6/Q.823 – Corrélation de nom pour les fragments élément géré, performance, état et administration

7 Description du modèle d'information

Ce paragraphe offre une description informelle de haut niveau du modèle d'information de gestion de trafic.

Le sous-paragraphe 7.1 contient une brève description de chaque classe d'objets utilisée dans le modèle d'information et fournit les informations suivantes:

- but de la classe d'objets;
- attributs de la classe d'objets;
- relation entre la classe d'objets et les autres classes d'objets.

Les attributs communs à plusieurs classes d'objets sont décrits en 7.2.

Le sous-paragraphe 7.3 décrit les actions qui agissent sur plusieurs classes d'objets du modèle d'information.

Le sous-paragraphe 7.4 décrit les aspects communs des notifications utilisées dans le modèle d'information.

Dans les tableaux énumérant les attributs des classes d'objets, les attributs hérités du sommet "Recommandation X.721" ne sont pas mentionnés explicitement, bien qu'ils soient présents dans ces objets par héritage.

7.1 Description de la classe d'objets

Ce paragraphe est subdivisé en sous-paragraphe dans lesquels les classes d'objets du modèle d'information sont décrites, dans la mesure où elles ne sont pas définies dans d'autres Recommandations. Pour les classes d'objets définies dans d'autres Recommandations, les références appropriées sont spécifiées.

7.1.1 Fragment élément géré

Ce fragment fournit une définition des ressources associées au commutateur.

7.1.1.1 Indicateur du niveau d'encombrement *congestionLevelIndication*

Une instance de la classe *congestionLevelIndication* fournit une indication du niveau d'encombrement courant de l'instance d'objet d'élément géré dans laquelle l'instance de cette classe est implantée. Le système d'exploitation n'est pas autorisé à instancier cette classe. Voir le Tableau 1.

Tableau 1/Q.823 – Entité *congestionLevelIndication*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
congestionLevelIndicationId	monovalué	obligatoire
congestionLevel	monovalué	obligatoire

Les attributs suivants définissent l'entité *congestionLevelIndication*:

- *congestionLevelIndicationID* – identificateur d'indication du niveau d'encombrement: utilisé comme RDN
- *congestionLevel* – niveau d'encombrement: c'est-à-dire encombrement courant d'un commutateur. Il est exprimé par les niveaux d'encombrement de machine (MCL, *machine congestion levels*) suivants:
 - MCL0: pas d'encombrement du commutateur;
 - MCL1: encombrement modéré du commutateur; celui-ci continue à travailler. Certains appels peuvent être rejetés;
 - MCL2: niveau d'encombrement élevé, le commutateur n'est plus capable de traiter tout le trafic offert. Il est rejeté plus d'appels qu'au niveau MLC1;
 - MCL3: incapable de traiter les appels. Bien qu'il soit souhaitable qu'ils le fassent, certains éléments de réseau peuvent être incapables de fournir un niveau 3 lors de pannes catastrophiques.

7.1.1.2 Destination difficile d'accès *htrDestination*

Une instance de l'objet *htrDestination* représente une destination qui est identifiée comme difficile d'accès. Il appartient au système d'exploitation ou de service de gestion des ressources du commutateur de décider si une destination est facile ou difficile d'accès; cette décision est prise suite à une information extérieure (par exemple tremblement de terre) ou en fonction du taux de tentatives de prise avec réponse ou du taux de prises avec réponse. L'attribut *administrativeState* (état administratif) permet de verrouiller l'état difficile d'accès de façon que temporairement il n'en soit tenu aucun compte. L'état HTR d'une destination peut aussi être corrélé avec les sous-faisceaux de circuits. Si l'attribut *administrativeState* est vide alors la destination est considérée comme HTR pour tous les sous-faisceaux de circuits possibles.

Une telle instance peut être créée explicitement par le gestionnaire (en cas de destination HTR établie en mode manuel) ou automatiquement par l'élément de réseau (en cas de détermination automatique par l'agent).

Dans ce modèle, le mécanisme de reconnaissance de l'état difficile ou facile d'accès par le système de gestion des ressources du commutateur est local: il n'est donc pas modélisé.

Une destination qui est inhibée (*administrativeState* = verrouillé) ou pour laquelle aucun objet *htrDestination* n'est instancié doit être considérée comme facile d'accès.

Toutes les instances de la classe d'objets *htrDestination* forment la liste HTR. Voir le Tableau 2.

Tableau 2/Q.823 – Entité *htrDestination*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>htrDestinationId</i>	monovalué	obligatoire
<i>creatorIdentity</i>	monovalué	obligatoire
<i>destinationType</i>	monovalué	conditionnel
<i>destinationCode</i>	monovalué	obligatoire
<i>administrativeState</i>	monovalué	obligatoire
<i>tmCircuitEndPointSubgroupList</i>	multivalué	conditionnel

Les attributs suivants définissent l'entité *htrDestination*:

- *htrDestinationId* – identificateur de destination HTR: utilisé comme RDN.
- *creatorIdentity* – identité du créateur de l'instance d'objet.
- *destinationType* – type de destination: nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs de type énuméré.
- *destinationCode* – indicatif de destination: c'est-à-dire indicatif de pays ou de zone ou code d'identification d'un commutateur ou numéro d'un autre lieu.
- *administrativeState* – état administratif: indique si l'indicatif *destinationCode* a été inhibé à partir de l'état HTR. Si cet attribut a la valeur "locked" alors l'indicatif est inhibé, s'il a la valeur "unlocked" alors l'indicatif n'est pas inhibé. La valeur par défaut est "unlocked".
- *tmCircuitEndPointSubgroupList* – liste de sous-faisceau de circuits surveillés dans le cadre de la gestion de trafic: liste de sous-faisceaux de circuits pour lesquels l'indicatif *destinationCode* est HTR.

7.1.1.3 Élément géré

La classe d'objets gérés *managedElement* est définie dans la Recommandation M.3100.

7.1.1.4 Sous-faisceau d'extrémité de circuits dans le cadre de la gestion du trafic *tmCircuitEndPointSubgroup*

La classe *tmCircuitEndPointSubgroup* est une sous-classe de la classe *circuitEndPointSubgroup* définie dans la Recommandation M.3100. Elle est utilisée pour l'élaboration des commandes et la surveillance des performances à des fins de gestion du trafic. Voir le Tableau 3.

Tableau 3/Q.823 – Entité *tmCircuitEndPointSubgroup*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
tmSurveillance	monovalué	obligatoire

L'attribut suivant définit l'entité *tmCircuitEndPointSubgroup*:

- *tmSurveillance* – l'attribut surveillance de la gestion de trafic indique si on est en train de surveiller le sous-faisceau *tmCircuitEndPointSubgroup* à des fins de gestion du trafic. Si l'attribut a la valeur Vrai (TRUE), l'élément de réseauinstanciera automatiquement la classe de données courantes *tmCircuitEndPointSubgroupCurrentData* correspondante. S'il a la valeur Faux (FALSE), l'élément de réseau supprimera cette instance.

7.1.2 Fragment surveillance des performances

Afin d'exécuter les fonctions de gestion du trafic, des données de performances seront périodiquement collectées, en particulier les ressources gérées suivantes: instances de sous-faisceau de circuits, de destination, de commutateur et de commande de trafic. Les données de performances seront collectées au moyen du modèle d'information Q.822 (données *currentData* ou données chronologiques *historyData*, par exemple) et du releveur simple *simpleScanner* décrit dans la Recommandation X.738. L'Appendice V fournit un guide d'instructions sur la manière dont ces données peuvent être recueillies avec le mécanisme de rapports émis sur demande ou de façon autonome.

7.1.2.1 Données courantes *currentData*

La classe d'objets gérés *currentData* est définie dans la Recommandation Q.822. La présente Recommandation décrit les sous-classes qui existent pour les différentes entités dont les attributs à surveiller sont définis. Dans la présente Recommandation, un ensemble minimal d'attributs est défini. Dans le cadre d'une réalisation technologique spécifique ou d'une réalisation particulière qui requiert plus d'attributs que ceux de l'ensemble minimal, il est recommandé d'organiser la classe *currentData* en sous-classes.

7.1.2.2 Données courantes de sous-faisceau d'extrémité de circuits *CircuitEndPointSubgroupCurrentData*

La classe *CircuitEndPointSubgroupCurrentData* est une sous-classe de la classe *currentData*. Elle sert à surveiller les données de performance des sous-faisceaux de circuits conformément à la définition de la Recommandation E.502.

Les attributs de surveillance des performances des sous-faisceaux de circuits dépendent de la directionnalité des sous-faisceaux; ces derniers peuvent être entrants, sortants ou bidirectionnels.

A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou toute autre classe qui en dérive et qui pourrait être instanciée. Voir le Tableau 4.

Tableau 4/Q.823 – Entité *circuitEndPointSubgroupCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
incomingSeizures	monovalué	obligatoire
outgoingBids	monovalué	obligatoire
outgoingSeizures	monovalué	obligatoire
answeredOutgoingSeizures	monovalué	obligatoire
answeredIncomingSeizures	monovalué	optionnel
overflow	monovalué	obligatoire
incomingTrafficUsage	monovalué	obligatoire
outgoingTrafficUsage	monovalué	obligatoire
numberOfAvailCircuits	monovalué	obligatoire

Les attributs suivants définissent l'entité *circuitEndPointSubgroupCurrentData*:

- *incomingSeizures* – nombre de prises entrantes du sous-faisceau de circuits [type 21 b) du 4.2/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits entrants ou bidirectionnels.
- *outgoingBids* – nombre de tentatives de prises sortantes c'est-à-dire de tentatives réussies ou manquées pour prendre un circuit du sous-faisceau de circuits [type 21 a) du 4.2/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits entrants ou bidirectionnels. Il ne tient pas compte des tentatives survenant avant l'application de toute commande de gestion du réseau.
- *outgoingSeizures* – nombre de prises sortantes c'est-à-dire de tentatives sortantes réussies pour obtenir un circuit du sous-faisceau de circuits [type 21 b) du 4.2.6/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels.

- *answeredOutgoingSeizures* – nombre de prises sortantes avec réponse c'est-à-dire pour lesquelles un signal de réponse a été reçu [type 21 c) du 4.2/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels.
- *answeredIncomingSeizures* – nombre de prises entrantes avec réponse c'est-à-dire pour lesquelles un signal de réponse a été émis vers l'arrière en direction du commutateur précédent. Ce nombre est valide pour les sous-faisceaux de circuits entrants ou bidirectionnels.
- *overflow* (débordement) – nombre de tentatives sortantes qui débordent de ce sous-faisceau de circuits [type 21 d) du 4.2.6/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels. Il ne tient pas compte des appels affectés par les commandes annulation du trafic de débordement réacheminé (*cancel rerouted overflow*), débordement temporaire depuis (*TAR from*), annulation du débordement depuis (*CancelFrom*).
- *incomingTrafficUsage* – trafic entrant écoulé [type 21 e) du 4.2.6/E.502]. Ce nombre, exprimé en erlang, est valide pour les sous-faisceaux de circuits entrants ou bidirectionnels. D'ordinaire, il est déterminé avec une méthode d'échantillonnage.
- *outgoingTrafficUsage* – trafic sortant écoulé [type 21 e) du 4.2.6/E.502]. Ce nombre, exprimé en erlang, est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels. D'ordinaire, il est déterminé avec une méthode d'échantillonnage.
- *numberOfAvailCircuits* – nombre de circuits disponibles c'est-à-dire capables d'écouler du trafic y compris ceux qui sont en train d'en écouler [type 21 f) du 4.2.6/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants, entrants ou bidirectionnels. Le fait qu'il s'agisse d'une valeur instantanée ou moyenne est laissé à la discrétion du fabricant à cause des changements normalement très fréquents sur les circuits. Les deux méthodes sont équivalentes.

7.1.2.3 Données courantes de commutateurs *exchangeCurrentData*

La classe *exchangeCurrentData* est une sous-classe de la classe d'objets *currentData*. Elle sert à surveiller les données de performance du commutateur conformément à la définition de la Recommandation E.502.

A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou toute autre classe qui en dérive.

Les attributs de surveillance des performances du commutateur dépendent des caractéristiques de flux des principaux types de trafic montrés sur la Figure 4/E.502. Voir le Tableau 5.

Tableau 5/Q.823 – Entité *exchangeCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
incomingTraffic	monovalué	obligatoire
outgoingTraffic	monovalué	obligatoire
transitTraffic	monovalué	obligatoire
terminatingTraffic	monovalué	obligatoire
originatingTraffic	monovalué	obligatoire
internalTraffic	monovalué	obligatoire
callsBlockedByLoadShedding	monovalué	obligatoire

Les attributs suivants définissent l'entité *exchangeCurrentData*:

- *incomingTraffic* – trafic entrant: c'est-à-dire nombre d'appels entrants dans le commutateur pour lesquels il a été accusé réception d'au moins un numéro [flux de trafic B de la Figure 4/E.502, types 4 a) et 20 a) du 4.2/E.502 et Appendice I].
- *outgoingTraffic* – trafic sortant: c'est-à-dire nombre d'appels sortants (prises) du commutateur qui ont abouti à une prise de circuit [flux de trafic R de la Figure 4/E.502, type 10 a) du 4.2/E.502, Appendice I].
- *transitTraffic* – trafic de transit: c'est-à-dire nombre d'appels de transit (prises) qui sont écoulés dans le commutateur [flux de trafic L de la Figure 4/E.502, type 6 a) du 4.2/E.502].
- *terminatingTraffic* – le trafic d'arrivée: c'est-à-dire nombre d'appels (prises) qui se présentent sur la chaîne d'arrivée du commutateur – à laquelle sont raccordés les joncteurs d'abonnés – [flux de trafic Q de la Figure 4/E.502, type 12 d) du 4.2/E.502].
- *originatingTraffic* – trafic de départ: c'est-à-dire nombre d'appels (prises) qui se présentent sur la chaîne de départ du commutateur [flux de trafic A de la Figure 4/E.502, type 1 a) du 4.2/E.502].
- *internalTraffic* – trafic interne: c'est-à-dire nombre d'appels internes (prises) que le commutateur établit [flux de trafic F de la Figure 4/E.502, type 2 a) du 4.2/E.502].
- *callsBlockedByLoadShedding* – trafic bloqué par un mécanisme de délestage: c'est-à-dire nombre d'appels qui ne peuvent pas être traités à cause d'un mécanisme de protection interne au commutateur contre la surcharge [type 20 g) du 4.2/E.502].

7.1.2.4 Données courantes de destination observée *observedDestinationCurrentData*

La classe *observedDestinationCurrentData* est une sous-classe de la classe d'objets *CurrentData*. Elle sert à surveiller les performances de destination conformément à la définition de la Recommandation E.502.

A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou une sous-classe qui en dérive. Voir le Tableau 6.

Tableau 6/Q.823 – Entité *observedDestinationCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
bids	monovalué	obligatoire
outgoingSeizures	monovalué	obligatoire
answeredOutgoingSeizures	monovalué	obligatoire
noCircuitsAvailable	monovalué	obligatoire
callsAffectedByDcc	monovalué	optionnel

Les attributs suivants définissent l'entité *observedDestinationCurrentData*:

- *bids* – nombre de tentatives de prises réussies pour la destination observée [type 22 a) du 4.2/E.502].
- *outgoingSeizures* – nombre de tentatives de prises sortantes: c'est-à-dire réussies pour la destination observée [type 22 b) du 4.2/E.502].
- *answeredOutgoingSeizures* – nombre de prises sortantes avec réponse de la destination observée pour lesquelles un signal de prise a été reçu [type 22 c) du 4.2/E.502].

- *noCircuitsAvailable* (aucun circuit disponible) – nombre de tentatives aboutissant à des communications inefficaces parce qu'il n'existe aucun circuit libre et disponible vers la destination observée; c'est-à-dire que l'on a un débordement sur le dernier sous-faisceau de circuits correspondant à cette destination. En général, *noCircuitsAvailable* \leq (*bids* – *outgoingSeizures*) [type 22 d) du 4.2.6/E.502].
- *callsAffectedByDcc* – appels affectés par la commande d'indicatif de destination: nombre d'appels vers la destination observée qui ont été affectés par la fonction de commande d'indicatif de destination [type 22 e) du 4.2/E.502].

7.1.2.5 Données courantes de commande de trafic *trafficControlCurrentData*

La classe d'objets *trafficControlCurrentData* est une sous-classe de la classe d'objets *currentData* définie dans la Recommandation Q.822.

Elle sert à surveiller l'efficacité d'une commande de trafic conformément à la définition de la Recommandation E.502.

Cette classe doit être indiquée pour chaque instance de commande. S'il est nécessaire de la préciser avec la classe d'objets de commande, ou d'indiquer l'incidence d'une commande sur une entité gérée, il est possible de synthétiser ces informations au niveau du système d'exploitation à partir de l'instance de commande correspondante.

A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou toute autre sous-classe qui en dérive. Voir le Tableau 7.

Tableau 7/Q.823 – Entité *trafficControlCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>callsOfferedToTrafficControl</i>	monovalué	optionnel
<i>callsAffectedByTrafficControl</i>	monovalué	obligatoire

Les attributs suivants définissent l'entité *trafficControlCurrentData*:

- *callsOfferedToTrafficControl* – nombre d'appels offerts à l'instance de commande de trafic.
- *callsAffectedByTrafficControl* – nombre d'appels couramment affectés par l'instance de commande de trafic.

7.1.2.6 Données courantes de sous-faisceau d'extrémité de circuits dans le cadre de la gestion du trafic *tmCircuitEndPointSubgroupCurrentData*

La classe d'objets *tmCircuitEndPointSubgroupCurrentData* est une sous-classe de la classe d'objets *CircuitEndPointSubgroupCurrentData*.

Elle sert à surveiller les données de performance des sous-faisceaux de circuits conformément à la définition de la Recommandation E.502 dans le cadre de la gestion du trafic. Cet objet ou ses sous-classes sont instanciés à des fins de gestion du trafic.

Une instance d'un objet de cette classe ne générera qu'une instance d'un objet de la classe *tmCircuitEndPointSubgroupHistoryData* (Données chronologiques de sous-faisceaux d'extrémités de circuits dans le cadre de la gestion du trafic).

Toutes les instances d'objet de la classe *tmCircuitEndPointSubgroupCurrentData* à l'intérieur du même élément géré auront la même période *granularityPeriod*.

Afin de synchroniser cette période avec la période de temps globale suivante après avoir instancié les sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de synchronisation de période *periodSynchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*). Voir le Tableau 8.

Tableau 8/Q.823 – Entité *tmCircuitEndPointSubgroupCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
historyRetention	monovalué	obligatoire

Les attributs suivants définissent l'entité *tmCircuitEndPointSubgroupCurrentData*:

- *historyRetention* – conservation de chronologie: définie dans la Recommandation Q.822. A des fins de gestion du trafic, sa valeur permise est égale à "1".

7.1.2.7 Données courantes de performance de commutateur dans le cadre de la gestion du trafic *tmExchangeCurrentData*

La classe d'objets *tmExchangeCurrentData* est une sous-classe de la classe d'objets *ExchangeCurrentData*.

Elle sert à surveiller les données de performance de commutateur conformément à la définition de la Recommandation E.502 dans le cadre de la gestion de trafic. Cette classe d'objets ou ses sous-classes sont instanciées à des fins de gestion de trafic.

Une instance d'un objet de cette classe ne générera qu'une instance d'objet de la classe *tmExchangePerformanceHistoryData*.

Afin de synchroniser la période granulaire *granularityPeriod* avec la période de temps globale suivante après avoir instancié les sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de synchronisation de période *periodSynchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*). Voir le Tableau 9.

Tableau 9/Q.823 – Entité *tmExchangeCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
historyRetention	monovalué	obligatoire

Les attributs suivants définissent l'entité *tmExchangeCurrentData*:

- *historyRetention* – conservation de chronologie: définie dans la Recommandation Q.822. A des fins de gestion du trafic, sa valeur permise est égale à "1".

7.1.2.8 Données courantes de destination observées dans le cadre de la gestion du trafic (*tmObservedDestinationCurrentData*)

La classe *tmObservedDestinationCurrentData* est une sous-classe de la classe d'objets *observedDestinationCurrentData*.

Elle sert à surveiller les données de performance d'une destination conformément à la définition de la Recommandation E.502 dans le cadre de la gestion de trafic. Cette classe d'objets ou ses sous-classes sont instanciées à des fins de gestion du trafic.

Une instance d'un objet de cette classe ne générera qu'une instance d'objet de la classe *tmObservedDestinationHistoryData*.

Toutes les instances d'objet *tmDestinationCurrentData* à l'intérieur du même élément géré auront la même période granulaire.

Afin de synchroniser la période granulaire *granularityPeriod* avec la période de temps globale suivante après l'instance des sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de synchronisation de période *periodSynchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*). Voir le Tableau 10.

Tableau 10/Q.823 – Entité *tmObservedDestinationCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
historyRetention	monovalué	obligatoire

L'attribut suivant définit l'entité *tmObservedDestinationCurrentData*:

- *historyRetention* – conservation de chronologie: définie dans la Recommandation Q.822. A des fins de gestion du trafic, sa valeur permise est égale à "1".

7.1.2.9 Données courantes de commande de trafic dans le cadre de la gestion du trafic *tmTrafficControlCurrentData*

La classe *tmTrafficControlCurrentData* est une sous-classe de la classe d'objets *TrafficControlCurrentData*.

Elle sert à surveiller l'efficacité des commandes de trafic conformément à la définition de la Recommandation E.502 dans le cadre de la gestion de trafic. Cette classe d'objets ou ses sous-classes sont instanciées à des fins de gestion du trafic.

Une instance d'un objet *tmTrafficControlCurrentData* de cette classe ne générera qu'une instance d'objet de la classe *tmTrafficControlHistoryData*. Toutes les instances d'objet *tmTrafficControlCurrentData* à l'intérieur du même élément géré auront la même période granulaire.

Afin de synchroniser la période granulaire *granularityPeriod* avec la période de temps globale suivante après l'instance des sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de synchronisation de période *periodSynchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*). Voir le Tableau 11.

Tableau 11/Q.823 – Entité *tmTrafficControlCurrentData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
historyRetention	monovalué	obligatoire

L'attribut suivant définit l'entité *tmTrafficControlCurrentData*:

- *historyRetention* – conservation de chronologie: définie dans la Recommandation Q.822. A des fins de gestion du trafic, sa valeur permise est égale à "1".

7.1.2.10 Données chronologiques *historyData*

La classe d'objets gérés *historyData* est définie dans la Recommandation Q.822.

Pour chacune des entités surveillées, les instances *historyData* contiennent une copie des attributs de performance qui sont présents dans les instances d'objets des sous-classes *currentData* correspondantes à la fin de la période granulaire.

Conformément aux définitions de la Recommandation Q.822, la présente Recommandation prend en charge deux types de corrélation de nom pour la classe *historyData*: corrélation avec la classe *currentData* et avec l'entité surveillée. Il est possible d'utiliser chacune de ces deux corrélations dans le cadre de la gestion du trafic, mais la corrélation de nom doit être systématiquement utilisée pour toutes les instances de la classe *historyData* à l'intérieur d'un système géré; c'est-à-dire que ces instances seront toutes nommées à partir de la classe *currentData* et de ses sous-classes ou à partir de l'entité surveillée et de ses sous-classes.

7.1.2.11 Données chronologiques de sous-faisceau d'extrémité de circuits *circuitEndPointSubgroupHistoryData*

La classe *circuitEndPointSubgroupHistoryData* dans le cadre de la gestion de trafic est une sous-classe de la classe d'objets *historyData*. Elle sert à surveiller les données de performance de sous-faisceau de circuits conformément à la définition de la Recommandation E.502.

Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante. Voir le Tableau 12.

Tableau 12/Q.823 – Entité *circuitEndPointSubgroupHistoryData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>incomingSeizures</i>	monovalué	obligatoire
<i>outgoingBids</i>	monovalué	obligatoire
<i>outgoingSeizures</i>	monovalué	obligatoire
<i>answeredOutgoingSeizures</i>	monovalué	obligatoire
<i>answeredIncomingSeizures</i>	monovalué	conditionnel
<i>overflow</i>	monovalué	obligatoire
<i>incomingTrafficUsage</i>	monovalué	obligatoire
<i>outgoingTrafficUsage</i>	monovalué	obligatoire
<i>numberOfAvailCircuits</i>	monovalué	obligatoire

Les attributs suivants définissent l'entité *circuitEndPointSubgroupHistoryData*:

- *incomingSeizures* – nombre de prises entrantes du sous-faisceau de circuits [type 21 b) du 4.2/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits entrants et bidirectionnels.
- *outgoingBids* – nombre de tentatives de prises sortantes: c'est-à-dire de tentatives réussies ou manquées pour prendre un circuit du sous-faisceau de circuits [type 21 a) du 4.2/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits entrants ou bidirectionnels. Il ne tient pas compte des tentatives survenant avant l'application de toute commande de gestion du réseau.

- *outgoingSeizures* – nombre de prises sortantes: c'est-à-dire de tentatives sortantes réussies pour obtenir un circuit du sous-faisceau de circuits [type 21 b) du 4.2.6/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels.
- *answeredOutgoingSeizures* – nombre de prises sortantes avec réponse: c'est-à-dire pour lesquelles un signal de réponse a été reçu [type 21 c) du 4.2.6/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels.
- *answeredIncomingSeizures* – nombre de prises entrantes avec réponse: c'est-à-dire pour lesquelles un signal de réponse a été émis vers l'arrière en direction du commutateur précédent. Ce nombre est valide pour les sous-faisceaux de circuits entrants ou bidirectionnels.
- *overflow* – débordement: nombre de tentatives sortantes qui débordent de ce sous-faisceau de circuits [type 21 d) du 4.2.6/E.502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels. Il ne tient pas compte des appels affectés par les commandes annulation du trafic de débordement réacheminé *cancel rerouted overflow*, débordement temporaire depuis (*tar from*), annulation du débordement depuis *cancel from*.
- *incomingTrafficUsage* – trafic entrant écoulé [type 21 e) du 4.2.6/E.502]. Ce nombre, exprimé en erlang, est valide pour les sous-faisceaux de circuits entrants ou bidirectionnels. D'ordinaire il est déterminé avec une méthode d'échantillonnage.
- *outgoingTrafficUsage* – trafic sortant écoulé [type 21 e) du 4.2.6/E.502]. Ce nombre, exprimé en erlang, est valide pour les sous-faisceaux de circuits sortants ou bidirectionnels. D'ordinaire, il est déterminé avec une méthode d'échantillonnage.
- *numberOfAvailCircuits* – nombre de circuits disponibles: c'est-à-dire capables d'écouler du trafic y compris ceux qui sont en train d'en écouler [type 21 f) du 4.2.6/502]. Ce nombre est valide pour les sous-faisceaux de circuits sortants, entrants ou bidirectionnels. Le fait qu'il s'agisse d'une valeur instantanée ou moyenne est laissé à la discrétion du fabricant à cause des changements normalement très fréquents sur les circuits. Les deux méthodes sont équivalentes.

7.1.2.12 Données chronologiques de commutateur *exchangeHistoryData*

La classe *exchangeHistoryData* est une sous-classe de la classe objet *historyData*. Elle sert à surveiller les données de performance de commutateur conformément à la définition de la Recommandation E.502.

Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante.

Les attributs de surveillance des performances d'un commutateur dépendent des caractéristiques de flux des principaux types de trafic montrés sur la Figure 4/E.502. Voir le Tableau 13.

Tableau 13/Q.823 – Entité *exchangeHistoryData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>incomingTraffic</i>	monovalué	obligatoire
<i>outgoingTraffic</i>	monovalué	obligatoire
<i>transitTraffic</i>	monovalué	obligatoire
<i>terminatingTraffic</i>	monovalué	obligatoire
<i>originatingTraffic</i>	monovalué	obligatoire
<i>internalTraffic</i>	monovalué	obligatoire
<i>callsBlockedByLoadShedding</i>	monovalué	obligatoire

Les attributs suivants définissent l'entité *exchangeHistoryData*:

- *incomingTraffic* – trafic entrant: c'est-à-dire nombre d'appels entrants dans le commutateur pour lesquels il a été accusé réception d'au moins un numéro [flux de trafic B de la Figure 4/E.502, types 4 a) et 20 a) du 4.2/E.502, Appendice I].
- *outgoingTraffic* – trafic sortant: c'est-à-dire nombre d'appels sortants (prises) du commutateur qui ont abouti à une prise de circuit [flux de trafic R de la Figure 4/E.502, type 10 a) du 4.2/E.502, Appendice I].
- *transitTraffic* – trafic de transit: c'est-à-dire nombre d'appels de transit (prises) qui sont écoulés dans le commutateur [flux de trafic L de la Figure 4/E.502, type 6 a) du 4.2/E.502].
- *terminatingTraffic* – trafic d'arrivée: c'est-à-dire nombre d'appels (prises) qui se présentent sur la chaîne d'arrivée du commutateur – à laquelle sont raccordés les joncteurs d'abonnés – [flux de trafic Q de la Figure 4/E.502, type 12 d) du 4.2/E.502].
- *originatingTraffic* – trafic de départ: c'est-à-dire nombre d'appels (prises) qui se présentent sur la chaîne de départ du commutateur [flux de trafic A de la Figure 4/E.502, type 1 a) du 4.2/E.502].
- *internalTraffic* – trafic interne: c'est-à-dire nombre d'appels internes (prises) que le commutateur établit [flux de trafic F de la Figure 4/E.502, type 2 a) du 4.2/E.502].
- *callsBlockedByLoadShedding* – trafic bloqué par un mécanisme de délestage: c'est-à-dire nombre d'appels qui ne peuvent pas être traités à cause du mécanisme de protection interne au commutateur contre la surcharge [type 20 g) du 4.2/E.502].

7.1.2.13 Destination observée *observedDestination*

L'objet *observedDestination* est instancié quand une destination doit être surveillée à des fins de gestion des performances, par exemple pour déterminer si elle est difficile d'accès. La destination est un pays, une zone, un commutateur ou un autre emplacement dans lequel le demandé est situé; elle peut être spécifiée à l'intérieur du pays. Elle est définie à l'Annexe A/E.410.

Une destination peut aussi être observée de façon sélective sur un ensemble de sous-faisceaux de circuits. Si l'objet *tmCircuitEndPointSubgroupList* n'est pas présent, la destination sera observée pour tous les sous-faisceaux de circuits. Voir le Tableau 14.

Tableau 14/Q.823 – Entité *observedDestination*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>observedDestinationId</i>	monovalué	obligatoire
<i>creatorIdentity</i>	monovalué	optionnel
<i>destinationType</i>	monovalué	conditionnel
<i>destinationCode</i>	monovalué	obligatoire
<i>tmCircuitEndPointSubgroupList</i>	multivalué	conditionnel
<i>tmSurveillance</i>	monovalué	obligatoire

Les attributs suivants définissent l'entité *observedDestination*:

- *observedDestinationId* – identificateur de destination observée: sert comme RDN.
- *creatorIdentity* – identité du créateur de l'instance d'objet.
- *destinationType* – type de destination: nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs de type énuméré.
- *destinationCode* – indicatif de destination: indicatif de pays ou/et de zone ou/et code d'identification d'un commutateur ou/et numéro d'un autre lieu.
- *tmCircuitEndPointSubgroupList* – liste de sous-faisceaux de circuits surveillés dans le cadre de la gestion du trafic: liste de sous-faisceaux de circuits pour lesquels la destination sera observée.
- *tmSurveillance* – surveillance de la gestion de trafic: indique si on est en train de surveiller la gestion de trafic. Si cet attribut a la valeur Vrai, l'élément de réseau instancie automatiquement la classe *tmObservedDestinationCurrentData*. S'il a la valeur Faux, l'élément de réseau supprime cette instance.

7.1.2.14 Données chronologiques de la destination observée *observedDestinationHistoryData*

La classe *observedDestinationHistoryData* est une sous-classe de la classe d'objets *historyData*. Elle sert à surveiller les données de performance de destination conformément à la définition de la Recommandation E.502.

Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante. Voir le Tableau 15.

Tableau 15/Q.823 – Entité *observedDestinationHistoryData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>bids</i>	monovalué	obligatoire
<i>outgoingSeizures</i>	monovalué	obligatoire
<i>answeredOutgoingSeizures</i>	monovalué	obligatoire
<i>noCircuitsAvailable</i>	monovalué	obligatoire
<i>callsAffectedByDcc</i>	monovalué	conditionnel

Les attributs suivants définissent l'entité *observedDestinationHistoryData*:

- *bids* – nombre de tentatives de prises pour la destination observée [type 22 a) du 4.2.6/E.502].
- *outgoingSeizures* – nombre de tentatives de prises sortantes: c'est-à-dire réussies pour la destination observée [type 22 b) du 4.2.6/E.502].
- *answeredOutgoingSeizures* – nombre de prises sortantes avec réponse pour la destination observée et pour lesquelles un signal de prise a été reçu [type 22 c) du 4.2.6/E.502].
- *noCircuitsAvailable* – aucun circuit disponible: nombre de tentatives aboutissant à des communications inefficaces parce qu'il n'existe aucun circuit libre et disponible pour la destination observée; c'est-à-dire que l'on a un débordement sur le dernier sous-faisceau de circuits correspondant à cette destination [type 22 d) du 4.2.6/E.502].
- *callsAffectedByDcc* – nombre d'appels vers la destination observée qui ont été affectés par les commandes de trafic liées aux destinations [type 22 e) du 4.2/E.502].

7.1.2.15 Données chronologiques de commande de trafic *trafficControlHistoryData*

La classe *trafficControlHistoryData* est une sous-classe de la classe d'objets *historyData* définie dans la Recommandation Q.822.

Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante. Voir le Tableau 16.

Tableau 16/Q.823 – Entité *trafficControlHistoryData*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>callsOfferedToTrafficControl</i>	monovalué	conditionnel
<i>callsAffectedByTrafficControl</i>	monovalué	obligatoire

Les attributs suivants définissent l'entité *trafficControlHistoryData*:

- *callsOfferedToTrafficControl* – nombre d'appels offerts à l'instance de commande de trafic.
- *callsAffectedByTrafficControl* – nombre d'appels couramment affectés par l'instance de commande de trafic.

7.1.3 Fragment commande

Les commandes de gestion du trafic servent à assurer une utilisation efficace de la capacité du réseau et à maintenir des performances satisfaisantes, afin de faire face aux demandes fluctuantes du trafic et aux situations d'urgence. Elles peuvent être déclenchées par une entrée spécifique à partir d'un système d'exploitation ou automatiquement en réponse à un stimulus interne ou externe.

Les données de performance du trafic sur le réseau sont cruciales à la fois pour les commandes automatiques et manuelles, afin de savoir quand appliquer ou supprimer ces commandes. Par ailleurs, pour évaluer les effets d'une commande sur la capacité et les performances d'écoulement du trafic sur le réseau, on doit obtenir des données qui fournissent ces informations. Des données de performances du trafic doivent donc être définies pour chaque entité commandable (destination, sous-faisceau de circuits, par exemple); une capacité de taxation doit aussi exister pour chaque commande active, afin d'en mesurer les effets. Une telle capacité est modélisée par un objet géré de surveillance des performances.

La présente Recommandation modélise les commandes automatiques et manuelles du trafic.

Commandes manuelles

Les commandes manuelles sont activées et supprimées par des actions de gestion spécifiques effectuées par un système d'exploitation. Dans ce modèle, l'activation et la suppression des commandes manuelles sont modélisées par la création et la suppression d'un objet géré "commande". De tels objets ne peuvent pas être mis à l'état désactivé; il est toutefois possible de positionner leurs paramètres sur des valeurs qui n'influenceront pas sur le volume normal du trafic ou ses caractéristiques d'acheminement (le pourcentage d'appels à bloquer pourrait par exemple être mis à 0).

Commandes automatiques

Les commandes automatiques sont modélisées par un mécanisme de déclenchement et un mécanisme de commande. Le mécanisme de déclenchement définit les conditions dans lesquelles une commande doit être appliquée; il émet automatiquement les indications au moment de l'activation de la commande. Le mécanisme de commande, qui est préassigné, spécifie la nature et l'effet des commandes à appliquer en réponse à une indication du mécanisme de déclenchement. Les deux mécanismes sont modélisés comme des objets gérés. Le mécanisme de commande est pré réglé dans l'élément de réseau par le gestionnaire de réseau. L'objet "déclenchement" peut aussi être créé par le gestionnaire de réseau. Le mécanisme de déclenchement est toujours situé dans l'élément de réseau. Les deux mécanismes peuvent être situés dans le même élément de réseau (commande ADC de type décentralisé, par exemple) ou dans deux éléments de réseau différents (commande ADC de type centralisé, par exemple).

Il est possible d'activer ou de désactiver les commandes automatiques et leurs mécanismes de déclenchement. Ces commandes sont activées ou désactivées par les indications de déclenchement reçues. Il est possible de les inhiber ou de les activer ainsi que leur mécanisme de déclenchement en réponse aux actions du système d'exploitation sur la gestion du trafic en positionnant l'attribut *administrativeState* respectivement sur "locked" ou "unlocked".

Si un mécanisme de déclenchement est verrouillé, il n'émet pas d'indication lorsque la condition de déclenchement est Vrai. Ceci provoque l'activation de toute nouvelle commande automatique (dans l'élément de réseau considéré ou tout autre élément de réseau).

Si une commande automatique est positionnée sur "locked", la commande ne sera pas activée même si l'indication de déclenchement est reçue.

Pour autoriser le système de gestion à avoir un contrôle total sur les commandes automatiques, deux attributs d'état ont été introduits dans tous les objets de "commande" pouvant être activés automatiquement. L'attribut *autoActivated* (auto-activé) indique si un mécanisme de déclenchement de l'activation des commandes est en cours de fonctionnement. L'attribut *administrativeState* indique si la commande a été verrouillée par le gestionnaire. Une commande automatique ne sera active que si l'attribut *autoActivated* a la valeur Vrai et si l'attribut *administrativeState* a la valeur déverrouillé. Si une commande est verrouillée lors de la réception de l'indication de déclenchement, elle passera à l'état auto-activé mais ne sera pas active à moins que ou tant que le gestionnaire n'aura pas déverrouillé l'attribut *administrativeState*.

L'Appendice III montre des diagrammes de transition d'état pour lesquels les commandes manuelles et automatiques sont activées ou désactivées.

7.1.3.1 Réduction automatique de l'encombrement ACC

La réduction automatique de l'encombrement est une commande préassignée, qui est activée automatiquement, lorsque le signal de déclenchement de l'encombrement est reçu du commutateur adjacent. Cette commande est définie au 4.1/E.412. Dans les commutateurs qui ont reçu l'indication d'encombrement, tous les sous-faisceaux qui se terminent au niveau du commutateur encombré doivent être soumis aux commandes de trafic qui limiteront la quantité de trafic envoyé à ce commutateur. Les types de traitements pouvant être utilisés pour étrangler le trafic sont annulation et évitement.

La commande ACC dépend d'un temporisateur automatique que l'élément de réseau règle de façon interne. Lorsque l'indicateur d'encombrement est reçu, un temporisateur démarre dans le commutateur qui a reçu cette indication. Les indicateurs subséquents redémarrent le temporisateur. Si aucun indicateur subséquent n'est reçu durant la temporisation, la commande est automatiquement désactivée. La valeur de temporisation est choisie par le fabricant et n'est pas gérable par le système d'exploitation. Les aspects liés à la temporisation ACC sont décrits au 5.5.2 b)/Q.542.

Si l'on souhaite traiter différemment les appels pour chaque niveau d'encombrement, un objet ACC distinct doit être instancié. Par ailleurs, les instances ACC, dont le niveau d'encombrement et l'attribut *controlTmCircuitEndPointSubgroup* sont les mêmes, doivent utiliser le même attribut *dispositionOfCall* (traitement des appels).

Une instance de la classe *assocAccAffectedTraffic* (trafic régulé ACC), vers laquelle pointe l'instance *assocAccAffectedTraffic* (trafic régulé ACC composite) doit avoir lieu avant l'instance de la classe ACC.

La commande ACC est protectrice. Sa classe est une sous-classe de la classe *trafficControl*. Voir le Tableau 17.

Tableau 17/Q.823 – Entité ACC

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>dispositionOfCalls</i>	monovalué	obligatoire
<i>administrativeState</i>	monovalué	obligatoire
<i>autoActivated</i>	monovalué	obligatoire
<i>assocAccAffectedTraffic</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité ACC:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau d'extrémité de circuits auquel s'applique cette commande.
- *dispositionOfCalls* – traitement des appels: identifie si les appels sont déviés vers le prochain sous-faisceau disponible (si la valeur est Néant) ou annulés (si la valeur est traitement).
- *administrativeState* – état administratif: identifie si le gestionnaire de réseau a verrouillé la commande. Si la valeur est positionnée sur "locked", la commande ne sera pas activée. Si elle est sur "unlocked", la commande sera activée. Lorsque la commande est créée, sa valeur est sur "unlocked". Les seules valeurs autorisées pour cet attribut sont "locked" et "unlocked".

- *autoActivated* – auto-activé: indique si un mécanisme de déclenchement de cette commande est actif. Si cet attribut a la valeur "TRUE" et si l'attribut *administrativeState* a la valeur "unlocked", la commande automatique est activée. Cette commande est désactivée pour toutes les autres combinaisons de valeurs des attributs *autoActivated* et *administrativeState*. Un changement de valeur de cet attribut ne générera pas automatiquement d'avis de changement de valeur d'attribut.
- *assocAccAffectedTraffic* – trafic régulé ACC composite: pointe vers l'instance d'objet *AccAffectedTraffic* correspondant aux niveaux d'encombrement 1 et 2. Cet attribut doit être non nul.

7.1.3.2 Trafic régulé ACC *AccAffectedTraffic*

L'objet *AccAffectedTraffic* identifie l'effet de la commande qui correspond aux différents niveaux d'encombrement résultant d'une combinaison donnée des aspects liés à l'acheminement, la destination, l'origine des appels et/ou d'autres critères.

Les aspects liés à l'origine, à la destination et à l'acheminement de l'appel ainsi qu'à des critères supplémentaires constituent tous dans cet ordre des clés.

Si une instance *AccAffectedTraffic* est supprimée, toutes les instances d'objets ACC en rapport doivent l'être. Voir le Tableau 18.

Tableau 18/Q.823 – Entité *AccAffectedTraffic*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>accAffectedTrafficId</i>	monovalué	obligatoire
<i>cl1ResponseCategories</i>	monovalué	obligatoire
<i>cl2ResponseCategories</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *AccAffectedTraffic*:

- *accAffectedTrafficId* – identificateur de trafic régulé ACC: sert comme RDN.
- *cl1ResponseCategories* – l'attribut catégories de réponses ACC identifie dans une séquence l'effet, en pourcentage, qui correspond à une combinaison des aspects liés à l'acheminement, la destination, l'origine et/ou d'autres critères du niveau d'encombrement 1. Cet effet s'applique à la combinaison de l'un ou plusieurs de ces critères. Il est prévu de ne prendre en charge qu'un seul choix de pourcentage.
- *cl2ResponseCategories* – l'attribut catégories de réponses ACC identifie dans une séquence l'effet, en pourcentage, qui correspond à une combinaison des aspects liés à l'acheminement, la destination, l'origine et/ou d'autres critères du niveau d'encombrement 2. Cet effet s'applique à la combinaison de l'un et/ou plusieurs de ces critères. Il est prévu de ne prendre en charge qu'un seul choix de pourcentage.

7.1.3.3 Mécanisme de déclenchement ACC *accTrigger*

Le mécanisme *accTrigger* est situé dans le commutateur encombré. Quand il est à l'état déverrouillé, il émet des indications d'encombrement vers les commutateurs adjacents, pour qu'ils activent automatiquement la commande ACC préassignée, de façon à réduire le trafic vers le commutateur encombré. Quand ce mécanisme est à l'état verrouillé, le commutateur encombré n'émet pas d'indication d'encombrement vers les commutateurs adjacents.

Il est à noter que le niveau *congestionLevel* (niveau d'encombrement) est modélisé dans la classe d'objets *congestionLevelIndication* (indication du niveau d'encombrement). Voir le Tableau 19.

Tableau 19/Q.823 – Entité *accTrigger*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
accTriggerId	monovalué	obligatoire
administrativeState	monovalué	obligatoire

Les attributs suivants décrivent l'entité *accTrigger*:

- *accTriggerId* – identificateur de déclenchement ACC: sert comme RDN.
- *administrativeState* – état administratif: détermine si le commutateur encombré émet l'indication d'encombrement vers les commutateurs adjacents. Si cet attribut a la valeur "unlocked", l'indication sera émise; s'il a la valeur "locked", l'indication ne sera pas émise. Seules les valeurs "locked" et "unlocked" sont valides pour cet attribut.

7.1.3.4 Commande automatique de destination ADC

Il existe deux types de commande automatique de destination: les types centralisé et décentralisé.

Dans une commande ADC de type centralisé, l'encombrement concernant un indicatif de destination donné est détecté au niveau du nœud d'arrivée, lorsque le taux d'arrivée d'appels dépasse le seuil fixé pour cet indicatif. Cette commande est décrite au 4.3/E.412. Le déclenchement ADC identifie le dépassement du seuil, ce qui aboutit à l'envoi d'une indication à chaque nœud de départ, via des messages de signalisation indiquant que le trafic offert à cette destination doit être réduit. De tels messages spécifient les détails de la commande ainsi que sa valeur de temporisation. Dès que les nœuds les reçoivent, ils instancient l'objet ADC.

Dans une commande ADC de type décentralisé, l'encombrement d'une destination est détecté au niveau de l'élément de réseau de départ, en observant les tentatives de prises et les réponses pour un indicatif de destination choisi. Cette commande est décrite au 4.3/E.412. L'objet *adcTrigger* identifie la destination à surveiller, la valeur du seuil ainsi que l'effet de la commande. Quand le seuil est franchi au niveau de l'élément de réseau de départ, l'objet ADC est automatiquement instancié. Si la commande ADC est verrouillée, elle sera inactive même si le mécanisme de déclenchement est actif. La commande ADC ne sera active que si l'attribut *administrativeState* est à l'état déverrouillé.

Le type de commande ADC est identifié par l'attribut *adcType* qui peut prendre la valeur "centralized" ou "decentralized".

Cet objet de commande n'est instancié que par l'élément de réseau et non par le système d'exploitation lorsqu'il reçoit un message du nœud d'arrivée (type centralisé) ou lorsque le seuil est franchi au niveau de l'élément de réseau de départ (type décentralisé).

Le type et l'indicatif de destination constituent dans cet ordre des clés de la commande.

Lorsque l'objet commande est instancié, seul l'un des attributs d'effet suivants doit être présent: *adcPercentage* (pourcentage d'appels annulés suite à l'activation d'une commande ADC), *adcContinuousTimer* (temporisateur ADC continu), *adcAsynchronousTimer* (temporisateur ADC asynchrone) ou *adcLeakyBucket* (compteur à fuite ADC). Cette instance d'objet ADC est supprimée à l'expiration de la temporisation. Si un déclenchement subséquent est reçu avant l'expiration du temporisateur, les valeurs de l'instance ADC sont remplacées par les valeurs correspondant au dernier déclenchement.

La classe de cette commande est une sous-classe de la classe *trafficControl*. Voir le Tableau 20.

Tableau 20/Q.823 – Entité ADC

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>adcType</i>	monovalué	obligatoire
<i>destinationType</i>	monovalué	conditionnel
<i>destinationCode</i>	monovalué	obligatoire
<i>adcPercentage</i>	monovalué	conditionnel
<i>adcContinuousTimer</i>	monovalué	conditionnel
<i>adcAsynchronousTimer</i>	monovalué	conditionnel
<i>adcLeakyBucket</i>	monovalué	conditionnel
<i>treatment</i>	monovalué	obligatoire
<i>timeExpiration</i>	monovalué	obligatoire
<i>administrativeState</i>	monovalué	obligatoire
<i>autoActivated</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *ADC*:

- *adcType* – type de commande ADC: peut prendre les valeurs "centralized" ou "decentralized".
- *destinationType* – type de destination: nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs de type énuméré.
- *destinationCode* – indicatif d'une destination encombrée.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande. Quand cette commande est active, le pourcentage d'appels spécifiés est bloqué (annulé).
- *continuousTimer* – temporisateur continu: comporte le nombre d'appels et une temporisation. Une fois ce nombre de tentatives d'appels traités sur un cycle du temporisateur, aucune autre tentative n'est autorisée avant l'expiration de la temporisation (5 appels en 60 secondes, par exemple).
- *asynchronousTimer* – temporisateur asynchrone: spécifie un temps. Ce temporisateur est activé lorsque la tentative d'appel est autorisée; aucune autre tentative n'est autorisée avant son expiration.
- *leakyBucket* – compteur à fuite: comporte la taille du compteur (valeur maximale autorisée pour le compteur) et le pas de décrémentation par unité de temps. Si la valeur de comptage dépasse la taille maximale définie, la tentative d'appel est annulée. Si cette valeur est inférieure à la taille maximale, la tentative est autorisée et le compteur est incrémenté. Celui-ci est décrément à intervalles de temps définis ce qui rend l'acceptation de nouveaux appels possible.
- *treatment* – traitement: fait à l'appel quand ce dernier est annulé suite à une commande (annonce, par exemple). Le traitement est déterminé par l'élément de réseau quand l'objet *commande* est instancié.

- *timeExpiration* – intervalle de temps après lequel la commande sera automatiquement supprimée: agit comme une hystérésis afin d'éviter les fréquentes oscillations entre l'instance d'une commande et la suppression de cette instance.
- *administrativeState* – état administratif: détermine si le gestionnaire de réseau a inhibé la commande. Si cet attribut a la valeur "locked", la commande ne sera pas activée. S'il a la valeur "unlocked", la commande sera activée. Les valeurs "locked" et "unlocked" sont les seules valides dans la définition de cette classe.
- *autoActivated* – détermine si le mécanisme de déclenchement de cette commande est actif. Puisque la commande n'est instanciée que lorsque le mécanisme est actif, la valeur de cet attribut est toujours "TRUE". La valeur "FALSE" est invalide dans ce cas.

7.1.3.5 Mécanisme de déclenchement de la commande automatique de destination *adcTrigger*

L'objet *adcTrigger* est configuré par le système d'exploitation avec les attributs nécessaires de façon à être automatiquement instancié lorsque le seuil de déclenchement est franchi (dans l'élément de réseau où se situe l'objet *adcTrigger* ou dans un élément de réseau adjacent). Le système d'exploitation peut spécifier la destination à surveiller par l'élément de réseau pour connaître l'encombrement, la valeur du seuil de déclenchement, l'effet de la commande et la temporisation.

L'objet *adcTrigger* peut être de type "centralized", "decentralized" ou "both" (centralisé/décentralisé) selon l'attribut *adcType*.

Si cet attribut est de type "centralized", le mécanisme *adcTrigger* et la commande ADC sont situés dans deux éléments de réseau différents. Un message de signalisation est envoyé de l'élément de réseau de départ (où se situe l'objet *adcTrigger*) vers tous les éléments de réseau adjacents afin qu'ils instancient la commande ADC.

Si l'attribut *adcType* est de type "decentralized", le mécanisme *adcTrigger* et la commande ADC sont tous deux situés dans l'élément de réseau de départ.

Si l'attribut *adcType* est de type "both", l'élément de réseau de départ (où se situe le mécanisme *adcTrigger*) envoie un message de signalisation à tous les éléments de réseau adjacents (c'est-à-dire comme pour le type "centralized") afin qu'ils instancient une commande ADC. Il instancie aussi la commande ADC en lui-même (c'est-à-dire comme pour le type "decentralized").

Un système d'exploitation peut instancier l'objet *adcTrigger*, mais pas l'objet commande ADC. L'objet *adcTrigger* peut aussi être mis à l'état verrouillé. Quand il est dans cet état, il n'envoie pas automatiquement un message au commutateur adjacent et n'instancie pas la commande ADC même si le seuil de déclenchement est franchi.

Seul l'un des attributs d'effet (*percentage*, *continuousTimer*, *asynchronousTimer* ou *leakyBucket*) doit être présent lorsque l'objet commande est instancié.

Le type et l'indicatif de destination constituent dans cet ordre des clés de la commande. Voir le Tableau 21.

Tableau 21/Q.823 – Entité *adcTrigger*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>adcTriggerId</i>	monovalué	obligatoire
<i>adcTriggerType</i>	monovalué	obligatoire
<i>destinationType</i>	monovalué	conditionnel
<i>destinationCode</i>	monovalué	conditionnel
<i>triggerThreshold</i>	monovalué	conditionnel
<i>percentage</i>	monovalué	conditionnel
<i>continuousTimer</i>	monovalué	conditionnel
<i>asynchronousTimer</i>	monovalué	conditionnel
<i>leakyBucket</i>	monovalué	conditionnel
<i>administrativeState</i>	monovalué	obligatoire
<i>timeExpiration</i>	monovalué	obligatoire

Les attributs suivants définissent l'entité *adcTrigger*:

- *adcTriggerId* – identificateur de déclenchement ADC: sert comme RDN.
- *adcTriggerType* – type de l'objet *adcTrigger*.
- *destinationType* – type de destination: nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs de type énuméré.
- *destinationCode* – indicatif d'une destination à surveiller dans le cadre de l'encombrement.
- *triggerThreshold* – seuil de déclenchement. Si ce seuil est franchi, la commande ADC est instanciée.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande. Si cette commande est active, le pourcentage d'appels spécifiés est bloqué (annulé).
- *continuousTimer* – temporisateur continu: comporte le nombre d'appels et une temporisation. Une fois ce nombre de tentatives d'appels traités sur un cycle du temporisateur, aucune autre tentative n'est autorisée avant l'expiration de la temporisation (5 appels en 60 secondes, par exemple).
- *asynchronousTimer* – temporisateur asynchrone: spécifie un temps. Le temporisateur est activé lorsque la tentative d'appel est autorisée; aucune autre tentative n'est autorisée avant son expiration.
- *leakyBucket* – compteur à fuite: comporte la taille du compteur (valeur maximale autorisée pour le compteur) et le pas de décrémentation par unité de temps. Si la valeur de comptage dépasse la taille maximale définie, la tentative d'appel est annulée. Si cette valeur est inférieure à la taille maximale, la tentative est autorisée et le compteur est incrémenté. Celui-ci est décrément à intervalles de temps définis, ce qui rend l'acceptation de nouveaux appels possible.
- *administrativeState* – état administratif: détermine si le gestionnaire de réseau a verrouillé le mécanisme de déclenchement. Si cet attribut a la valeur "locked", la commande ADC ne sera pas instanciée. S'il a la valeur "unlocked", la commande sera instanciée.
- *timeExpiration* – expiration: intervalle de temps après lequel la commande sera automatiquement supprimée.

7.1.3.6 Annulation du débordement depuis *cancelFrom*

Cette commande est activée manuellement sur un sous-faisceau sortant et empêche le trafic de déborder sur les sous-faisceaux suivants de la chaîne. Elle est définie au 3.2.1/E.412.

Cette commande est protectrice et s'applique aux sous-faisceaux de circuits. Sa classe est une sous-classe de la classe *trafficControl*.

Elle peut servir à empêcher le débordement des trafics de premier choix et de débordement. L'Appendice IV en explique le principe. La distinction entre les deux types de trafic se fait en conférant à l'attribut *RoutingAspect* (aspect lié à l'acheminement) le type de trafic approprié.

L'attribut *controlTmCircuitEndPointSubgroup* constitue la clé de priorité la plus élevée, suivi par chaque composante des attributs *originationAspect* (aspect lié à l'origine), *destinationAspect* (aspect lié à la destination) et *routingAspect*. La valeur la plus spécifique est prioritaire. Voir le Tableau 22.

Tableau 22/Q.823 – Entité *cancelFrom*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>routingAspect</i>	monovalué	obligatoire
<i>destinationAspect</i>	monovalué	obligatoire
<i>OriginationAspect</i>	monovalué	obligatoire
<i>percentage</i>	monovalué	obligatoire
<i>treatment</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *cancelFrom*:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau d'extrémité de circuit auquel s'applique cette instance d'objet commande.
- *routingAspect* – aspect lié à l'acheminement (trafic de premier choix ou de débordement) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic s'applique à tous les types de trafic.
- *destinationAspect* – aspect lié à la destination (HTR, par exemple) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic est valide pour tous les aspects liés à la destination.
- *originationAspect* – catégorie de l'appelant et l'origine (conformément à la Recommandation Q.763) pour lesquelles cette commande est valide. Si la valeur de cet attribut est une séquence vide, la commande de trafic est valide pour tous les aspects liés à l'origine.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande. Si cette commande est active, le pourcentage d'appels spécifiés est bloqué (annulé).
- *treatment* – traitement fait à l'appel quand il est annulé suite à une commande (annonce, par exemple).

7.1.3.7 Annulation du trafic de débordement réacheminé *cancelRerouted*

Cette commande empêche le réacheminement ou le débordement du trafic qui a déjà été réacheminé. Le trafic réacheminé n'est pas autorisé à déborder sur le sous-faisceau auquel s'applique la commande *cancelRerouted*, tandis que le trafic de débordement normal n'en est pas affecté. Cette commande est définie au 3.2.4/E.412.

Il est à noter que pour appliquer cette commande, chaque appel doit être marqué par un message de signalisation chaque fois qu'une commande de débordement temporaire TAR lui est appliquée.

Cette commande est manuelle et protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*. Voir le Tableau 23.

Tableau 23/Q.823 – Entité *cancelRerouted*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>treatment</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *cancelRerouted*:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau de circuits auquel l'instance de cet objet commande s'applique.
- *treatment* – traitement fait au flux de trafic annulé (annonce, par exemple).

7.1.3.8 Annulation du débordement vers *cancelTo*

Cette commande concerne l'annulation des trafics de premier choix et de débordement vers un sous-faisceau *controlTmCircuitEndPointSubgroup* donné.

La commande d'annulation du trafic de premier choix empêche ce trafic d'accéder à un sous-faisceau de circuits. Elle est définie au 3.1.2/E.412.

La commande d'annulation du trafic de débordement est activée sur un sous-faisceau sortant et empêche ce trafic d'accéder au sous-faisceau auquel la commande s'applique. Elle est définie au 3.2.1/E.412.

La distinction entre ces deux types de trafic se fait en attribuant à l'attribut *routingAspect* une valeur appropriée.

L'Appendice IV explique comment empêcher les trafics de premier choix et de débordement d'accéder à un sous-faisceau en utilisant la commande *cancelTo*.

L'attribut *controlTmCircuitEndPointSubgroup* constitue la clé de priorité la plus élevée, suivi par chacune des composantes des attributs *originationAspect*, *destinationAspect* et *routingAspect*. La valeur la plus spécifique est prioritaire.

Cette commande est manuelle et protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*. Voir le Tableau 24.

Tableau 24/Q.823 – Entité *cancelTo*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>routingAspect</i>	monovalué	obligatoire
<i>destinationAspect</i>	monovalué	obligatoire
<i>originationAspect</i>	monovalué	obligatoire
<i>percentage</i>	monovalué	obligatoire
<i>treatment</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *cancelTo*:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau de circuits auquel l'instance de cet objet commande s'applique.
- *routingAspect* – aspect lié à l'acheminement (trafic de premier choix ou de débordement) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic s'applique à tous les types de trafic.
- *destinationAspect* – aspect lié à la destination (HTR, par exemple) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic est valide pour tous les types de destination.
- *originationAspect* – catégorie de l'appelant et origine (conformément à la Recommandation Q.763) pour lesquelles cette commande est valide. Si cet attribut est une séquence Néant, la commande de trafic est valide pour tous les aspects liés à l'origine.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande. Si cette commande est active, le pourcentage d'appels spécifiés est bloqué (annulé).
- *treatment* – traitement fait à l'appel quand il est annulé suite à une commande (annonce, par exemple)

7.1.3.9 Commande d'indicatif de destination *destinationCodeControl*

Cette commande manuelle interdit l'acheminement vers une destination donnée selon un pourcentage (se rapporter à l'intitulé blocage sur indicatif de la Recommandation E.412) ou un nombre d'appels (se rapporter à l'intitulé espacement d'appels de la Recommandation E.412).

Elle peut s'appliquer à un indicatif de pays ou de zone ou à un code d'identification de commutateur ou au numéro d'un autre lieu. Des précisions sont données au 3.1.1.1/E.412.

Lorsque cette commande est active en mode blocage sur indicatif, le pourcentage spécifié d'appels est bloqué (annulation).

Lorsqu'elle est active en mode espacement d'appels, le nombre d'appels libérés pour cette destination est commandé. Il peut être spécifié par l'un des trois attributs suivants: *continuousTimer*, *asynchronousTimer* ou *leakyBucket*.

L'effet de cette commande (blocage sur indicatif ou espacement d'appels) peut être défini à l'intérieur de la même classe ou dans une classe séparée. Le mode blocage sur indicatif s'applique lorsque l'effet ne concerne qu'une seule destination. Dans ce cas, il existe quatre attributs (*percentage*, *continuousTimer*, *asynchronousTimer* ou *leakyBucket*) qui spécifient cet effet dans la même classe; seul l'un d'entre eux doit être spécifié. Dans ce scénario, l'instance d'un seul objet agit comme une fonction de commande d'indicatif.

Le mode espacement d'appels définit l'effet de cette commande dans une classe d'objets séparée lorsque celui-ci s'applique à tout un groupe de destinations (le compteur à fuite montre où l'utiliser). Dans ce cas, il est nécessaire que l'attribut *assocOwnerDccGroup* (groupe DCC associé) pointe vers la classe d'objets contenant l'effet de la commande. Dans ce scénario, deux classes d'objets représentent ensemble une fonction de commande d'indicatif.

Les règles suivantes sont valides indépendamment du nombre de classes d'objets utilisées pour représenter la destination de la fonction de commande d'indicatif:

- il est impossible de créer deux ou plusieurs instances d'objet *destinationCodeControl* avec les mêmes combinaisons de valeurs d'attributs clés: *destinationType*, *destinationCode*, et *originationAspect*;
- l'attribut *destinationCode* constitue la clé de priorité la plus élevée, suivi dans cet ordre de l'attribut *destinationType* (s'il est présent) et des composantes de l'attribut *originationAspect* dans cet ordre;
- lorsqu'il existe plus d'une commande avec des numéros qui se chevauchent, la commande la plus spécifique du même type s'applique à la place de la commande la moins spécifique, chacune d'elles pouvant affecter le même appel.

Si l'attribut *assocOwnerDccGroup* est spécifié, alors l'attribut *dccGroup* associé doit exister. Les instances de cette classe d'objets appartiennent au même groupe si et seulement si elles ont la même valeur d'attribut *assocOwnerDccGroup*.

La classe de cette commande est une sous-classe de la classe d'objets *trafficControl*. Voir le Tableau 25.

Tableau 25/Q.823 – Entité *destinationCodeControl*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>destinationType</i>	monovalué	conditionnel
<i>destinationCode</i>	monovalué	obligatoire
<i>originationAspect</i>	monovalué	obligatoire
<i>percentage</i>	monovalué	conditionnel
<i>continuousTimer</i>	monovalué	conditionnel
<i>asynchronousTimer</i>	monovalué	conditionnel
<i>leakyBucket</i>	monovalué	conditionnel
<i>assocOwnerDccGroup</i>	monovalué	conditionnel
<i>treatment</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *destinationCodeControl*:

- *destinationType* – type de destination – nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs énumérées.
- *destinationCode* – indicatif de destination: indicatif de pays ou de zone ou code d'identification d'un commutateur ou numéro d'un autre lieu.
- *originationAspect* – catégorie de l'appelant et origine (conformément à la Recommandation Q.763) pour lesquelles cette commande est valide. Si cet attribut est une séquence vide, la commande de trafic est valide pour tous les aspects liés à l'origine.

- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande. Si cette commande est active, le pourcentage d'appels spécifiés est bloqué (annulé). Cet attribut ne sert que si l'effet de la commande se trouve dans la même classe d'objets que la commande elle-même.
- *continuousTimer* – temporisateur continu: comporte le nombre d'appels et une temporisation. Une fois ce nombre de tentatives d'appels traité sur un cycle du temporisateur, aucune autre tentative n'est autorisée avant l'expiration de la temporisation (5 appels en 60 secondes, par exemple). Cet attribut ne sert que si l'effet de la commande se trouve dans la même classe d'objets que la commande elle-même.
- *asynchronousTimer* – temporisateur asynchrone: spécifie un temps. Le temporisateur est activé lorsque la tentative d'appel est autorisée; aucune autre tentative n'est autorisée avant son expiration. Cet attribut ne sert que si l'effet de la commande se trouve dans la même classe d'objets que la commande elle-même.
- *leakyBucket* – compteur à fuite: comporte la taille du compteur (valeur maximale autorisée pour le compteur) et le pas de décrémentation par unité de temps. Si la valeur de comptage dépasse la taille maximale définie, la tentative d'appel est annulée. Si cette valeur est inférieure à la taille maximale, la tentative est autorisée et le compteur est incrémenté. Celui-ci est décrément à intervalles de temps définis ce qui rend l'acceptation de nouveaux appels possible. Cet attribut ne sert que si l'effet de la commande se trouve dans la même classe d'objets que la commande elle-même.
- *assocOwnerDccGroup* – L'attribut groupe DCC associé est présent si l'on doit appliquer le même effet à tout un groupe de destinations; c'est-à-dire, que cette instance *destinationCodeControl* est l'élément d'un groupe de commandes de destination auxquelles on doit appliquer le même effet. Cet attribut pointe vers l'instance d'objet *dccGroup* dans laquelle l'effet des commandes est défini.
- *treatment* – traitement fait à l'appel quand il est annulé suite à une commande (annonce, par exemple)

7.1.3.10 Groupe de commande d'indicatif de destination ou groupe DCC *dccGroup*

Cette classe d'objets définit l'effet de la commande d'indicatif de destination lorsqu'on souhaite appliquer le même effet à tout un groupe de destinations. Ses instances sont associées aux instances de la classe d'objets *destinationCodeControl* par une relation (1:n). Cette classe d'objets ne fournit pas par elle-même de fonction de commande d'indicatif de destination; quand une ou plusieurs de ses instances sont associées à une ou plusieurs des instances *destinationCodeControl*, elle représente la fonction de commande d'indicatif de destination.

Seul l'un des attributs d'effet (*percentage*, *continuousTimer*, *asynchronousTimer* ou *leakyBucket*) doit être présent lors de l'instance de l'objet commande. Voir le Tableau 26.

Tableau 26/Q.823 – Entité *dccGroup*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>dccGroupId</i>	monovalué	obligatoire
<i>percentage</i>	monovalué	conditionnel
<i>continuousTimer</i>	monovalué	conditionnel
<i>asynchronousTimer</i>	monovalué	conditionnel
<i>leakyBucket</i>	monovalué	conditionnel

Les attributs suivants décrivent l'entité *dccGroup*:

- *dccGroupId* – identificateur du groupe DCC: sert comme RDN.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande. Si cette commande est active, le pourcentage d'appels spécifiés est bloqué (annulé).
- *continuousTimer* – temporisateur continu: comporte le nombre d'appels et une temporisation. Une fois ce nombre de tentatives d'appels traité sur un cycle du temporisateur, aucune autre tentative n'est autorisée avant l'expiration de la temporisation (5 appels en 60 secondes, par exemple).
- *asynchronousTimer* – temporisateur asynchrone: spécifie un temps. Le temporisateur est activé lorsque la tentative d'appel est autorisée; aucune autre tentative n'est autorisée avant son expiration.
- *leakyBucket* – compteur à fuite: comporte la taille du compteur (valeur maximale autorisée pour le compteur) et le pas de décrémentation par unité de temps. Si la valeur de comptage dépasse la taille maximale définie, la tentative d'appel est annulée. Si cette valeur est inférieure à la taille maximale, la tentative est autorisée et le compteur est incrémenté. Celui-ci est décrémenté à intervalles de temps définis ce qui rend l'acceptation de nouveaux appels possible.

7.1.3.11 Commande de réservation sélective de circuit SCR

La commande SCR permet à un commutateur de donner automatiquement la préférence à certains attributs par rapport à d'autres (le trafic de premier choix sur le trafic de débordement, par exemple) lorsque les circuits sont encombrés ou sur le point de l'être. Cette commande peut être fournie avec un ou deux seuils et devenir ultérieurement plus sélective. Elle est préassignée avec des seuils fixant le nombre de circuits à laisser au repos selon les différents types de trafic offerts à un sous-faisceau. Si ce nombre de circuits au repos ou la capacité de réserve du sous-faisceau sont inférieurs ou égaux au seuil de réservation, l'appel est annulé ou dévié vers le sous-faisceau suivant de la chaîne. Cette commande est définie au 4.2/E.412.

La commande SCR se compose des paramètres suivants:

- seuils de réservation;
- réponse à une commande;
- traitement choisi.

Les seuils de réservation et les réponses des commandes, qui leur sont associées, sont déterminés par l'attribut *activationThresholds* (seuils d'activation) et l'instance d'objet *scrAffectedTraffic* (trafic réservé SCR) associée. Le traitement choisi pour les appels, dont l'accès au sous-faisceau a été refusé, est donné dans l'attribut *dispositionOfCalls* (traitement d'appels).

Si ce nombre de circuits au repos ou la capacité de réserve du sous-faisceau sont inférieurs ou égaux au seuil de réservation, le commutateur vérifiera l'instance d'objet *scrAffectedTraffic* indiquée, afin de déterminer si l'appel sera commandé. Les types de traitements pouvant servir à étrangler le trafic sont annulation et évitement.

Les niveaux définis dans l'attribut *activationThresholds* et la référence donnée aux attributs de trafic réservés SCR composite correspondront aux commandes à seuil unique ou multiple.

L'instance d'objet *scrAffectedTraffic* doit être présente avant l'instance de la classe d'objets SCR.

Cette commande est automatique et protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*. Voir le Tableau 27.

Tableau 27/Q.823 – Entité SCR

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>dispositionOfCalls</i>	monovalué	obligatoire
<i>administrativeState</i>	monovalué	obligatoire
<i>autoActivated</i>	monovalué	obligatoire
<i>activationThresholds</i>	monovalué	obligatoire
<i>assocScrAffectedTraffic</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité SCR:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau de circuits auquel s'applique la commande.
- *dispositionOfCalls* – traitement des appels: identifie si les appels sont déviés vers le prochain sous-faisceau de circuits disponible (si la valeur est Néant) ou annulés (si la valeur est traitement).
- *administrativeState* – état administratif: détermine si le gestionnaire de réseau a verrouillé la commande. Si l'attribut a la valeur "locked", la commande ne sera pas activée. S'il a la valeur "unlocked", la commande sera activée.
- *autoActivated* – auto-activé: indique si un mécanisme de déclenchement est actif pour cette commande. Si cet attribut a la valeur "TRUE" et si l'attribut *administrativeState* a la valeur "unlocked", la commande automatique est activée. Cette commande est désactivée pour toutes les autres combinaisons de valeurs des attributs *autoActivated* et *administrativeState*. Un changement de valeur de cet attribut ne générera pas automatiquement d'avis de changement de valeur d'attribut.
- *activationThresholds* – seuils d'activation: seuils de déclenchement de la commande.
- *assocScrAffectedTraffic* – pointe vers l'instance d'objet *scrAffectedTraffic* de niveau de réservation 1 et 2. Il doit être non nul.

7.1.3.12 Trafic réservé SCR *scrAffectedTraffic*

La classe *scrAffectedTraffic* représente la catégorie de réponse des commandes SCR.

Elle détermine l'effet de cette commande sur le trafic pour chacun des aspects suivants: acheminement, destination, origine ou autres critères.

Si l'attribut *level2ResponseCategories* Catégories de réponse de niveau 2 est spécifié, les attributs *activationThreshold* de niveaux 1 et 2 doivent être spécifiés avec la même unité (nombre ou pourcentage).

Les aspects liés à l'origine, la destination et l'acheminement et d'autres critères constituent dans cet ordre des clés. Voir le Tableau 28.

Tableau 28/Q.823 – Entité *scrAffectedTraffic*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>scrAffectedTrafficId</i>	monovalué	obligatoire
<i>level1ResponseCategories</i>	monovalué	obligatoire
<i>level2ResponseCategories</i>	monovalué	conditionnel

Les attributs suivants décrivent l'entité *scrAffectedTraffic*:

- *scrAffectedTrafficId* – identificateur de trafic réservé SCR: sert comme RDN.
- *level1ResponseCategories* – L'attribut catégories de réponse de niveau 1 identifie dans une séquence l'effet d'une combinaison des attributs *routingAspect* ou *destinationAspect*, *OriginationAspect* et tout autre critère supplémentaire de niveau de réservation 1.
- *level2ResponseCategories* – L'attribut catégories de réponse de niveau 2 identifie dans une séquence l'effet d'une combinaison des attributs *routingAspect* ou *destinationAspect*, *OriginationAspect* et tout autre critère supplémentaire de niveau de réservation 2.

7.1.3.13 Evitement *skip*

Cette commande est activée sur un faisceau de circuits sortant et sert à obliger le trafic à emprunter le prochain sous-faisceau de la table d'acheminement. Elle peut affecter les trafics de premier choix et de débordement. Elle est définie au 3.2.2/E.412.

L'attribut *controlTmCircuitEndPointSubgroup* constitue la clé de priorité la plus élevée, suivi de chaque composante des attributs *originationAspect*, *destinationAspect* et *routingAspect* dans cet ordre. La valeur la plus spécifique a la priorité.

Cette commande est manuelle et protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*. Voir le Tableau 29.

Tableau 29/Q.823 – Entité *skip*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/ Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>routingAspect</i>	monovalué	obligatoire
<i>destinationAspect</i>	monovalué	obligatoire
<i>originationAspect</i>	monovalué	obligatoire
<i>percentage</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *skip*:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau d'extrémité de circuit auquel s'applique cette instance d'objet commande de gestion du trafic.
- *routingAspect* – aspect lié à l'acheminement (trafic de premier choix ou de débordement) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic s'applique à tous les types de trafic.
- *destinationAspect* – aspect lié à la destination (HTR, par exemple) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic est valide pour tous les types de destination.

- *originationAspect* – catégorie de l'appelant et origine (conformément à la Recommandation Q.763) pour lesquelles cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic est valide pour tous les aspects liés à l'origine.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande.

7.1.3.14 Débordement temporaire depuis *tarFrom*

La commande *tarFrom* est une commande expansive qui augmente temporairement le nombre de possibilités d'acheminement afin d'établir des appels:

- par addition de nouveaux sous-faisceaux de circuits à la fin de la table d'acheminement;
- ou par insertion de nouveaux faisceaux de circuits entre les sous-faisceaux déjà existants de cette table,
afin de fournir des voies de débordement supplémentaires, qui sont normalement indisponibles dans la table.

Ces deux types de débordements temporaires (addition à la suite ou insertion entre des sous-faisceaux de circuits) sont décrits au 3.2.3/E.412. Cette commande ne s'applique qu'au niveau des sous-faisceaux de circuits et agit sur toutes les tables d'acheminement où l'attribut *controlTmCircuitEndPointSubgroup* apparaît. Les faisceaux de circuits de débordement temporaire doivent se terminer sur un commutateur qui a la capacité d'atteindre la destination finale.

L'effet de cette commande peut aussi être limité aux destinations HTR en positionnant l'attribut *destinationAspect* ou en spécifiant explicitement la destination avec l'attribut *destinationCode*. Si l'indicatif de destination est une chaîne vide, la commande s'applique aux destinations que ce sous-faisceau de circuits dessert.

Il est aussi possible de limiter l'effet de cette commande au trafic de premier choix ou de débordement.

Les attributs *controlTmCircuitEndPointSubgroup*, *destinationCode*, *destinationType* (s'il est présent), les composantes des attributs *originationAspect*, *destinationAspect* et *routingAspect* constituent dans cet ordre des clés de la commande.

Cette commande est manuelle et expansive. Sa classe est une sous-classe de la classe d'objets *trafficControl*. Voir le Tableau 30.

Tableau 30/Q.823 – Entité *tarFrom*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>newTmCircuitEndPointSubgroups</i>	monovalué	obligatoire
<i>routingAspect</i>	monovalué	obligatoire
<i>destinationAspect</i>	monovalué	obligatoire
<i>originationAspect</i>	monovalué	obligatoire
<i>percentage</i>	monovalué	obligatoire
<i>destinationType</i>	monovalué	conditionnel
<i>destinationCode</i>	monovalué	obligatoire
<i>returnAction</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *tarFrom*:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau d'extrémité de circuit auquel s'applique la commande *tarFrom*.
- *newTmCircuitEndPointSubgroups* – séquence de nouveaux faisceaux de circuits d'extrémité utilisés pour la gestion de trafic. Au moins un élément de la séquence doit être présent.
- *routingAspect* – aspect lié à l'acheminement (trafic de premier choix ou de débordement) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic s'applique à tous les types de trafic.
- *destinationAspect* – aspect lié à la destination (HTR, par exemple) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic est valide pour tous les types de destination.
- *originationAspect* – catégorie de l'appelant et origine (conformément à la Recommandation Q.763) pour lesquelles cette commande est valide. Si cet attribut est une séquence vide, la commande de trafic est valide pour tous les aspects liés à l'origine.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande.
- *destinationType* – type de destination – nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs énumérées.
- *destinationCode* – indicatif de destination: indicatif de pays ou de zone ou code d'identification d'un commutateur ou numéro d'un autre lieu. Si cet attribut est une chaîne vide, la commande est valide pour toutes les destinations du sous-faisceau *controlTmCircuitEndPointSubgroup*.
- *returnAction* – action en retour: traitement du trafic débordant des sous-faisceaux *newTmCircuitEndPointSubgroups*: il y a deux valeurs possibles; on peut:
 - positionner cet attribut sur la valeur Néant, ce qui provoque l'insertion de sous-faisceaux *newTmCircuitEndPointSubgroups* après le sous-faisceau *controlTmCircuitEndPointSubgroup*;
 - positionner cet attribut sur traitement, ce qui annulera le trafic qui a débordé des sous-faisceaux *newTmCircuitEndPointSubgroups*.

7.1.3.15 Débordement temporaire vers *tarTo*

La commande *tarTo* est une commande expansive qui augmente temporairement le nombre de possibilités d'acheminement pour établir les appels soit en ajoutant de nouveaux sous-faisceaux au début de la table d'acheminement de façon que le trafic soit d'abord offert à ces nouveaux sous-faisceaux, soit en remplaçant les sous-faisceaux de circuits existants par un ensemble de nouveaux sous-faisceaux de circuits. Un ou plusieurs faisceaux de circuits, qui sont normalement indisponibles dans la table d'acheminement normale mais qui sont au repos, sont rendus disponibles.

Ces deux types de débordement (addition avant ou remplacement de sous-faisceaux existants) sont décrits au 3.2.3/E.412. La commande *tarTo* ne s'applique qu'au niveau des sous-faisceaux de circuits et agit sur toutes les tables d'acheminement où le sous-faisceau *TmCircuitEndPointSubgroup* est présent. Les faisceaux de circuits de débordement temporaire doivent se terminer sur un commutateur qui a la capacité d'atteindre la destination finale.

L'effet de cette commande peut se limiter aux destinations de la liste HTR en positionnant l'attribut *routingAspect* ou en spécifiant explicitement la destination dans l'attribut *destinationCode*. Si ce dernier est une chaîne vide, la commande est valide pour toutes les destinations du sous-faisceau *TmCircuitEndPointSubgroup*.

Il est aussi possible de limiter l'effet de cette commande aux trafic de premier choix ou de débordement.

Les attributs *controlTmCircuitEndPointSubgroup*, *destinationCode*, *destinationType* (s'il est présent), les composantes des attributs *originationAspect*, *destinationAspect* et *routingAspect* constituent dans cet ordre des clés de la commande.

Cette commande est manuelle et expansive. Sa classe est une sous-classe de la classe *trafficControl*. Voir le Tableau 31.

Tableau 31/Q.823 – Entité *tarTo*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/Conditionnel
<i>controlTmCircuitEndPointSubgroup</i>	monovalué	obligatoire
<i>newTmCircuitEndPointSubgroups</i>	monovalué	obligatoire
<i>routingAspect</i>	monovalué	obligatoire
<i>destinationAspect</i>	monovalué	obligatoire
<i>originationAspect</i>	monovalué	obligatoire
<i>percentage</i>	monovalué	obligatoire
<i>destinationType</i>	monovalué	conditionnel
<i>destinationCode</i>	monovalué	obligatoire
<i>returnAction</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *tarTo*:

- *controlTmCircuitEndPointSubgroup* – sous-faisceau d'extrémité de circuit auquel s'applique la commande *tarTo*.
- *newTmCircuitEndPointSubgroups* – Séquence de nouveaux faisceaux de circuits d'extrémité utilisés pour la gestion de trafic. Au moins un élément de la séquence doit être présent.
- *routingAspect* – aspect lié à l'acheminement (trafic de premier choix ou de débordement) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant, la commande de trafic s'applique à tous les types de trafic.
- *destinationAspect* – aspect lié à la destination (HTR, par exemple) pour lequel cette commande est valide. Si la valeur de cet attribut est Néant la commande de trafic est valide pour tous les types de destination.
- *originationAspect* – catégorie de l'appelant et origine (conformément à la Recommandation Q.763) pour lesquelles cette commande est valide. Si cet attribut est une séquence vide, la commande de trafic est valide pour tous les *aspects liés à l'origine*.
- *percentage* – pourcentage d'appels annulés suite à l'activation d'une commande.
- *destinationType* – type de destination: nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs de type énuméré.
- *destinationCode* – indicatif de destination: indicatif de pays ou/et de zone ou/et code d'identification d'un commutateur ou/et numéro d'un autre lieu. Si la valeur de cet attribut est une chaîne vide, la commande est valide pour toutes les destinations du sous-faisceau *controlTmCircuitEndPointSubgroup*.

- *returnAction* – action en retour: traitement du trafic débordant des sous-faisceaux *newTmCircuitEndPointSubgroups*. Il y a trois valeurs possibles; on peut:
 - positionner cet attribut sur la valeur Néant ce qui provoque l'insertion de sous-faisceaux *newTmCircuitEndPointSubgroups* avant le sous-faisceau *controlTmCircuitEndPointSubgroup*;
 - positionner cet attribut sur évitement ce qui provoque le remplacement du sous-faisceau *controlTmCircuitEndPointSubgroup* par les sous-faisceaux *newTmCircuitEndPointSubgroups*.
 - positionner cet attribut sur traitement ce qui provoque le remplacement du sous-faisceau *controlTmCircuitEndPointSubgroup* et des sous-faisceaux de circuits restants du plan d'acheminement par les sous-faisceaux *newTmCircuitEndPointSubgroups*.

7.1.3.16 Commande de trafic *trafficControl*

La classe d'objets *trafficControl* est une "hyperclasse" pour toutes les classes d'objets représentant les commandes de trafic comme cela est défini dans la Recommandation E.412. Cette hyperclasse n'est pas instanciée.

Lorsque plusieurs commandes du même type sont instanciées dans la même ressource gérée, un mécanisme (auquel on se rapporte comme clé) est décrit dans chaque sous-classe de la classe *trafficControl* afin de choisir une instance de commande. La commande au critère le plus spécifique a la priorité.

Si plusieurs commandes sont actives, on appliquera la hiérarchie des commandes spécifiée dans la Recommandation E.412. En attendant d'introduire cette hiérarchie dans la Recommandation E.412, l'Appendice X indiquera la marche à suivre. Voir le Tableau 32.

Tableau 32/Q.823 – Entité *trafficControl*

Attributs	Monovalué/Multivalué	Optionnel/Obligatoire/Conditionnel
<i>trafficControlId</i>	monovalué	obligatoire
<i>creatorIdentity</i>	monovalué	optionnel
<i>tmSurveillance</i>	monovalué	obligatoire

Les attributs suivants décrivent l'entité *trafficControl*:

- *trafficControlId* – identificateur d'objet commande de trafic: sert comme RDN.
- *creatorIdentity* – identité du créateur de l'instance d'objet.
- *tmSurveillance* – surveillance de la gestion de trafic: indique si on est en train de surveiller l'instance *trafficControl*. Si cet attribut a la valeur Vrai, l'élément de réseau instancie automatiquement la classe *tmTrafficControlCurrentData* correspondante. S'il a la valeur Faux, l'élément de réseau supprime automatiquement cette instance.

7.1.4 Fragment état

Le fragment état vise à fournir un mécanisme permettant de recevoir un ensemble fini d'information d'état sur les différentes anomalies, les activités et les événements qui sont présents dans les éléments de réseaux et dont un résumé fourni à intervalles réguliers et rapprochés intéresse un gestionnaire de réseau.

7.1.4.1 Indicateur d'état *stateIndicator*

Cette classe d'objets définit les actions et les avis contenant un bit pour chaque condition. Si à tout moment lors de la période, la condition a été satisfaite, la valeur du bit est égale à 1; sinon sa valeur est égale à 0. On définit les bits suivants:

- a) *exchangeCongestionLevel1* – 1^{er} niveau d'encombrement du commutateur (CL1): ce bit est égal à 1 si durant la période *granularityPeriod*, l'élément de réseau se trouve au 1^{er} niveau d'encombrement;
- b) *exchangeCongestionLevel2* – 2^{ème} niveau d'encombrement du commutateur (CL2): ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau se trouve au 2^{ème} niveau d'encombrement;
- c) *congestionLevel1Received* – indicateur CL1 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL1 d'un des commutateurs adjacents;
- d) *congestionLevel2Received* – indicateur CL2 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL2 d'un des commutateurs adjacents;
- e) *scrTriggered* – déclenchement SCR: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur "unlocked" et si l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande SCR;
- f) *AccTriggered* – déclenchement ACC: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur "unlocked" et si l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande ACC;
- g) *protectiveControlActive* – commande protectrice activée: ce bit est égal à 1 si durant la période granulaire *granularityPeriod* une ou plusieurs instances de commandes protectrices manuelles (c'est-à-dire: *cancelTo*, *cancelFrom*, *skip*, *cancelRerouted*) sont présentes dans l'élément de réseau;
- h) *expansiveControlActive* – commande expansive activée: ce bit est égal à 1 si durant la période granulaire *granularityPeriod* une ou plusieurs instances de commandes expansives manuelles (c'est-à-dire: *tarTo*, *tarFrom*) sont présentes dans l'élément de réseau;
- i) *destinationControlActive* – activation de commande de destination: ce bit est égal à 1 si durant la période granulaire *granularityPeriod* une ou plusieurs instances de commandes de destination manuelles (c'est-à-dire: fonctions *destinationCodeControl*) sont présentes dans l'élément de réseau;
- j) *htrDestinationActive* – activation de commande de destination: ce bit est égal à 1 si durant la période granulaire *granularityPeriod* une ou plusieurs instances de commandes *htrDestination* sont présentes dans l'élément de réseau;
- k) *circuitEndPointSubgroupAddedOrDeleted* – addition ou suppression d'un sous-faisceau d'extrémité de circuits: ce bit est égal à 1 durant la période granulaire *granularityPeriod* si un sous-faisceau de circuits est ajouté ou supprimé dans l'élément de réseau;
- l) *accTransmissionInhibited* – transmission acc inhibée: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* du mécanisme *accTrigger* est à l'état verrouillé;
- m) *adcTriggered* – commande adc déclenchée: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* est à l'état déverrouillé et l'attribut *autoActivated* a la valeur VRAI pour au moins une instance de commande ADC.

La période granulaire *granularityPeriod* de cette classe sera inférieure ou égale à celle utilisée pour la classe *currentData* définie dans la présente Recommandation; elle sera spécifiée en seconde ou en minute.

7.1.5 Fragment Administratif

7.1.5.1 Discriminateur de retransmission d'événement *Event Forwarding Discriminator*

La classe d'objets gérés *EventForwardingDiscriminator* est définie dans la Recommandation X.721.

7.1.5.2 Releveur simple *SimpleScanner*

La classe d'objets gérés *SimpleScanner* est définie dans la Recommandation X.738.

7.2 Définition des attributs communs

Ce paragraphe décrit tous les attributs génériques qui sont utilisés dans ce modèle d'information. Les attributs suivants ont été identifiés et il est possible de trouver leur définition dans les normes appropriées, qui sont référencées dans la présente Recommandation:

- nom distinctif relatif;
- attributs d'état.

La définition des attributs propres à ce modèle d'information fait partie des définitions des classes d'objets.

7.2.1 Nom distinctif relatif

La sémantique de l'attribut RDN est spécifiée dans la Recommandation X.720. Ce type d'attribut sert à identifier une instance d'objet géré uniquement dans le contexte de son objet géré supérieur. Ses contraintes de syntaxe et de sémantique sont spécifiées dans la Recommandation X.720.

7.2.2 Attributs d'état

Dans ce modèle d'information, les attributs liés à l'état des objets gérés se composent des attributs du modèle d'états génériques conformément aux définitions de la Recommandation X.731 ainsi que de chaque attribut d'état qui est spécifié pour chacune des classes d'objets définies dans la présente Recommandation.

7.2.2.1 Etat administratif

La sémantique de l'attribut *administrativeState* est spécifiée dans la Recommandation X.731. Sa syntaxe est spécifiée dans la Recommandation X.721.

7.3 Description des actions

Toutes les actions décrites ci-dessous s'appliquent aux classes d'objets identifiées dans le tableau.

Action	Affected Object Class	Remarques
stateIndicatorAction	stateIndicator	néant

7.4 Descriptions des notifications

Le modèle d'information défini dans la présente Recommandation utilise les notifications génériques suivantes:

Notification	Recommandations de définition
notification de création d'objet	X.721 et X. 730
notification de suppression d'objet	X.721 et X. 730
notification de changement de valeur d'attribut	X.721 et X. 730
notification de changement d'état	X.721 et X. 731
notification de rapport de relève	X.738

Le modèle d'information défini dans la présente Recommandation utilise les notifications spécifiques suivantes:

Notification	Affected Objet Classe	Remarques
stateIndicatorNotification	stateIndicator	néant

8 Unités fonctionnelles

La présente Recommandation définit les unités fonctionnelles suivantes:

- A) unité fonctionnelle de commande de trafic: permet à un gestionnaire d'activer et de désactiver les commandes de trafic, et de modifier leurs attributs et leurs mécanismes de déclenchement (création/suppression/configuration des sous-classes de la classe *trafficControl*, tableaux de paramètres et déclenchement).
- B) unité fonctionnelle de mesures effectuées sur les commandes de trafic: permet à un gestionnaire de collecter les mesures effectuées sur les commandes de trafic.
- C) unité fonctionnelle de mesure du trafic dans les commutateurs: permet à un gestionnaire de collecter l'ensemble des mesures du trafic dans les commutateurs.
- D) unité fonctionnelle de mesure du trafic offert aux sous-faisceaux d'extrémité de circuits: permet à un gestionnaire de collecter les mesures de trafic offert aux sous-faisceaux d'extrémité de circuits.
- E) unité fonctionnelle de mesures effectuées sur les destinations: permet à un gestionnaire de collecter les mesures concernant les destinations.
- F) unité fonctionnelle d'accès aux données chronologiques: permet à un gestionnaire d'exécuter des opérations "GET" sur les sous-classes de données chronologiques.
- G) unité fonctionnelle d'identification des destinations difficiles d'accès: permet à un gestionnaire d'identifier les destinations difficiles d'accès (création/suppression/configuration des instances *htrDestination*).
- H) unité fonctionnelle d'accès aux données de référence: permet à un gestionnaire d'accéder, avec une opération "GET", aux instances de classes d'objets autres que les classes *historyData* ou *currentData*.
- I) unité fonctionnelle de notification d'indicateur d'état émis de façon autonome: requiert la prise en charge de la notification *stateIndicatorNotification* définie dans la présente Recommandation.

- J) unité fonctionnelle de notification d'indicateur d'état à la demande: requiert la prise en charge de l'action *stateIndicatorAction* définie dans la présente Recommandation.
- K) mesures d'activation et de désactivation: permet à un gestionnaire de spécifier les entités d'un certain type pour lesquelles des mesures doivent être collectées (positionnement de l'attribut *tmSurveillance*) ou la création d'une ou des sous-classes *currentData*.

Afin d'assurer l'interfonctionnement, le gestionnaire et l'agent doivent partager au moins une unité fonctionnelle commune et un ensemble commun de classes d'objets.

L'Appendice VI fournit un mappage des fonctions de gestion du trafic (énumérées à l'Appendice I) sur les unités fonctionnelles.

8.1 Unités fonctionnelles définies dans d'autres Recommandations

Au-delà des unités fonctionnelles énumérées ci-dessus, la présente Recommandation prend aussi en charge les unités fonctionnelles suivantes qui sont décrites dans les Recommandations du CCITT:

- i) unité fonctionnelle de stimulation de relevés (Recommandation X.738),
- ii) unité fonctionnelle de signalisation d'événement de récapitulation (Recommandation X.738),
- iii) unité fonctionnelle de gestion des rapports d'événement (Recommandation X.734).

8.2 Négociation des unités fonctionnelles

La présente Recommandation affecte la valeur suivante:

{joint-iso-ccitt ms(9) fonction Recommendation(0) q(17) q823(823) functionalUnitPackage(1)} à l'identificateur *FunctionalUnitPackageId* du type ASN.1 défini dans la Recommandation X.701, afin de négocier la disponibilité d'une des unités fonctionnelles suivantes:

- 0 unité fonctionnelle de commande du trafic
- 1 unité fonctionnelle de mesures effectuées sur les commandes de trafic
- 2 unité fonctionnelle de mesure du trafic dans les commutateurs
- 3 unité fonctionnelle de mesure du trafic offert aux sous-faisceaux d'extrémité de circuits
- 4 unité fonctionnelle de mesures effectuées sur la destination
- 5 unité fonctionnelle d'accès aux données chronologiques
- 6 unité fonctionnelle d'identification des destinations difficiles d'accès
- 7 unité fonctionnelle d'accès aux données de référence
- 8 unité fonctionnelle de notification d'indicateur d'état émis de façon autonome
- 9 unité fonctionnelle de notification d'indicateur à la demande
- 10 unité fonctionnelle d'activation et de désactivation des mesures

où le nombre identifie la position du bit dans la CHAÎNE BINAIRE attribuée aux unités fonctionnelles, les noms se rapportant à ces unités.

Dans le contexte d'application de la gestion des systèmes, le mécanisme de négociation des unités fonctionnelles est décrit dans la Recommandation X.701.

9 Relations avec les autres Recommandations

- La présente Recommandation utilise les définitions de la Recommandation X.721 "Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: définition des informations de gestion"

- La présente Recommandation utilise les services définis dans la Recommandation X.730 "Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion des objets"
- La présente Recommandation utilise les services définis dans la Recommandation X.731 "Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion d'états"
- La présente Recommandation utilise les définitions et les services de la Recommandation X.734 "*Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion des rapports d'événement*"
- La présente Recommandation utilise les définitions et les services de la Recommandation X.738 "Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de récapitulation"
- La présente Recommandation utilise les définitions et les services de la Recommandation Q.822 "Description d'étape 1, d'étape 2 et d'étape 3 de l'interface Q3 – Gestion de la qualité de fonctionnement"

10 Conformité

Les implémentations se déclarant conformes à la présente Recommandation doivent observer les prescriptions définies dans les sous-paragraphes suivants.

10.1 Conformité statique

Les implémentations seront conformes aux prescriptions de la présente Recommandation, qu'elles remplissent le rôle du gestionnaire, de l'agent ou les deux. La déclaration de conformité d'une implémentation sera conforme au moins à l'un de ces rôles, quand elle est disponible.

Pour prétendre à conformité avec le rôle du gestionnaire ou de l'agent, une implémentation prendra en charge au moins l'une des capacités suivantes:

- surveillance des performances;
- commande.

La surveillance des performances requiert au moins l'une des unités fonctionnelles énumérées aux points i), ii) ou F) et au moins l'une de celles énumérées aux points B), C), D) ou E). La capacité d'administration et de vérification peut optionnellement être incluse dans l'implémentation.

NOTE – Si la capacité de vérification est négociée, on peut envisager de négocier les unités fonctionnelles de sélection d'objet multiple et de filtrage définies dans la Recommandation X.710.

Capacités	Unités fonctionnelles	Classes d'objets
surveillance des performances	i), ii), F) B), C), D), E)	sous-classes d'objets de la classe <i>historyData</i> pour la surveillance (par exemple, <i>tmObservedDestinationHistoryData</i> , <i>tmExchangeHistoryData</i> , <i>tmTrafficControlHistoryData</i>)
application des commandes	A)	sous-classes d'objets des classes <i>trafficControl</i> (par exemple <i>acc</i> , <i>adc</i> , <i>cancelFrom</i>), <i>déclenchements</i> (par exemple <i>accTrigger</i>), et tableaux de paramètres (par exemple <i>accAffectedTraffic</i>)
vérification	i), H)	objets modélisant les ressources et les commandes (par exemple <i>circuitEndPointSubgroups</i> , <i>simpleScanner</i> , <i>cancelFrom</i>)
	I), J)	<i>stateIndicator</i>
administration		objets gérés de gestion du trafic du réseau
	iii)	discriminateur de retransmission d'événement
	(Note)	<i>simpleScanner</i>
	G)	<i>htrDestination</i>
NOTE – La Recommandation X.738 actuelle ne permet pas de gérer les sous-classes du releveur avec la granularité souhaitée pour la gestion du trafic. Il est donc nécessaire de négocier leur gestion au moment de la réalisation.		

Les objets gérés pris en charge par le système ouvert à des fins de gestion du trafic seront conformes à la syntaxe et à la sémantique du modèle d'information spécifié au paragraphe 9.

Cette implémentation prendra en charge la syntaxe de transfert dérivée des règles de codage spécifiées dans la Recommandation X.209 et nommées {joint-iso-ccitt asn1(1) basicEncoding(1)} pour les types abstraits de données référencés dans les définitions pour lesquelles la prise en charge est déclarée.

10.2 Conformité dynamique

Pour les rôles pour lesquels la conformité est déclarée, le système prendra en charge les éléments de procédures définis dans:

- Recommandation X.730 pour les services PT-GET, PT-CREATE, PT-DELETE, PT-SET;
- Recommandation X.730 pour les rapports de création et de suppression d'objet et de changement de valeur d'attribut si la classe d'objets appropriée est prise en charge;
- Recommandation X.731 pour le rapport de changement d'état si la classe d'objets appropriée est prise en charge.

10.3 Prescription en matière de déclaration de conformité de l'implémentation de gestion

Tous les formulaires MCS (récapitulatif de conformité de gestion), MICS (déclaration de conformité d'information de gestion), MOCS (formulaire de déclaration de conformité d'objet géré) et PICS (déclaration de conformité d'implémentation de protocole) conformes à la présente Recommandation seront techniquement identiques aux formulaires spécifiés dans les futures annexes de la présente Recommandation. Ils doivent maintenir la numérotation des tableaux et l'indexation des sujets, et ne modifier que la pagination et l'en-tête des pages.

Le fournisseur de l'implémentation prétendant à conformité avec la présente Recommandation remplira une copie du récapitulatif MCS (à fournir) qui fait partie des prescriptions de conformité ainsi que tous les autres formulaires ICS (déclaration de conformité de la réalisation) qui sont référencés comme applicables dans ce récapitulatif. Une déclaration ICS conforme à la présente Recommandation:

- décrira une implémentation conforme à la présente Recommandation;
- sera remplie conformément aux instructions données dans la Recommandation X.724;
- comportera l'information nécessaire pour identifier de façon unique le fournisseur et l'implémentation.

Les déclarations de conformité de l'information de gestion, qui sont définies dans la présente Recommandation et qui concernent les classes d'objets gérés définies ailleurs comporteront les prescriptions des formulaires MIDS (déclaration de définition d'information de gestion), comme cela sera spécifié dans les futures déclarations MOCS de la classe d'objets gérés.

11 Définitions formelles des classes d'objets

11.1 Définition des classes d'objets

11.1.1 ACC

acc MANAGED OBJECT CLASS

DERIVED FROM trafficControl;

CHARACTERIZED BY

"ITU-T Rec. M.3100":stateChangeNotificationPackage,
accPackage PACKAGE

BEHAVIOUR

accBehaviour BEHAVIOUR

DEFINED AS

"La réduction automatique de l'encombrement est une commande préassignée, qui est activée automatiquement, lorsque le signal de déclenchement de l'encombrement est reçu du commutateur adjacent. Cette commande est définie au 4.1/E.412. Dans les commutateurs qui ont reçu l'indication d'encombrement, tous les sous-faisceaux qui se terminent au niveau du commutateur encombré doivent être soumis aux commandes de trafic qui limiteront la quantité de trafic envoyé à ce commutateur. Les types de traitements pouvant être utilisés pour étrangler le trafic sont annulation et évitement. La commande ACC dépend d'un temporisateur automatique que l'élément de réseau règle de façon interne. Lorsque l'indicateur d'encombrement est reçu, un temporisateur démarre dans le commutateur qui a reçu cette indication. Les indicateurs subséquents redémarrent le temporisateur. Si aucun indicateur subséquent n'est reçu durant la temporisation, la commande est automatiquement désactivée. La valeur de temporisation est choisie par le fabricant et n'est pas gérable par le système d'exploitation. Les aspects liés à la temporisation ACC sont décrits au 5.5.2 b)/Q.542. Si l'on souhaite traiter différemment les appels pour chaque niveau d'encombrement, un objet ACC distinct doit être instancié. Par ailleurs, les instances ACC, dont le niveau d'encombrement et l'attribut *controlTmCircuitEndPointSubgroup* sont les mêmes, doivent utiliser le même attribut *dispositionOfCall*.

Les aspects liés à l'origine, à la destination et à l'acheminement de l'appel ainsi qu'à des critères supplémentaires constituent tous dans cet ordre des clés.

Une instance de la classe *assocAccAffectedTraffic*, vers laquelle pointe l'instance *assocAccAffectedTraffic*, doit avoir lieu avant l'instance de la classe ACC.

La commande ACC est protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*.";

ATTRIBUTES

controlTmCircuitEndPointSubgroup

GET,

dispositionOfCalls

GET-REPLACE,

"CCITT Rec. X.721:1992":administrativeState

DEFAULT VALUE Q823-TM-ASN1Module.defaultAdministrativeState

PERMITTED VALUES Q823-TM-ASN1Module.PermittedState

GET-REPLACE,

autoActivated

GET,

```
    assocAccAffectedTraffic          GET-REPLACE;
;;
REGISTERED AS {q823ObjectClass 1};
```

11.1.2 accAffectedTraffic

```
accAffectedTraffic MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721:1992":top;
CHARACTERIZED BY
    "ITU-T Rec. M.3100":objectManagementNotificationsPackage,
    accAffectedTrafficPackage PACKAGE
BEHAVIOUR
    accAffectedTrafficBehaviour BEHAVIOUR
DEFINED AS
    "L'objet accAffectedTraffic identifie l'effet de la commande qui correspond aux différents niveaux
    d'encombrement résultant d'une combinaison donnée des aspects liés à l'acheminement, la destination ou
    l'origine ou d'autres critères.
    Si une instance accAffectedTraffic est supprimée, toutes les instances d'objets ACC en rapport doivent
    l'être.";;
ATTRIBUTES
    accAffectedTrafficId          GET,
    cl1ResponseCategories         GET-REPLACE,
    cl2ResponseCategories         GET-REPLACE;
;;
REGISTERED AS {q823ObjectClass 2};
```

11.1.3 accTrigger

```
accTrigger MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721:1992":top;
CHARACTERIZED BY
    "ITU-T Rec. M.3100":stateChangeNotificationPackage,
    accTriggerPackage PACKAGE
BEHAVIOUR
    accTriggerBehaviour BEHAVIOUR
DEFINED AS
    "Le mécanisme accTrigger est situé dans le commutateur encombré. Quand il est à l'état déverrouillé, il
    émet des indications sur le niveau d'encombrement (congestionLevel) vers les commutateurs adjacents, pour
    qu'ils activent automatiquement la commande ACC préassignée, de façon à réduire le trafic vers le
    commutateur encombré. Quand ce mécanisme est à l'état verrouillé, le commutateur encombré n'émet pas
    d'indication sur le niveau d'encombrement vers les commutateurs adjacents. Il est à noter que ce niveau est
    modélisé dans la classe d'objets congestionLevelIndication";;
ATTRIBUTES
    accTriggerId          GET,
    "CCITT Rec. X.721:1992":administrativeState
        DEFAULT VALUE Q823-TM-ASN1Module.defaultAdministrativeState
        PERMITTED VALUES Q823-TM-ASN1Module.PermittedState
        GET-REPLACE;
;;
REGISTERED AS {q823ObjectClass 3};
```

11.1.4 adc

```
adc MANAGED OBJECT CLASS
DERIVED FROM trafficControl;
CHARACTERIZED BY
    "ITU-T Rec. M.3100":stateChangeNotificationPackage,
    adcPackage PACKAGE
```

BEHAVIOUR

adcBehaviour BEHAVIOUR

DEFINED AS

"Il existe deux types de commande automatique de destination: les types centralisé et décentralisé. Dans une commande ADC de type centralisé, l'encombrement concernant un indicatif de destination donné est détecté au niveau du nœud d'arrivée, lorsque le taux d'arrivée d'appels dépasse le seuil fixé pour cet indicatif. Cette commande est décrite au 4.3/E.412. Le déclenchement ADC identifie le dépassement du seuil, ce qui aboutit à l'envoi d'une indication à chaque nœud de départ, via des messages de signalisation indiquant que le trafic offert à cette destination doit être réduit. De tels messages spécifient les détails de la commande ainsi que sa valeur de temporisation. Dès que les nœuds les reçoivent, ils instancient l'objet ADC.

Dans une commande ADC de type décentralisé, l'encombrement d'une destination est détecté au niveau de l'élément de réseau de départ, en observant les tentatives de prises et les réponses pour un indicatif de destination choisi. Cette commande est décrite au 4.3/E.412. L'objet adcTrigger identifie la destination à surveiller, la valeur du seuil ainsi que l'effet de la commande. Quand le seuil est franchi au niveau de l'élément de réseau de départ, l'objet ADC est automatiquement instancié. Si la commande ADC est verrouillée, elle sera inactive même si le mécanisme de déclenchement est actif. La commande ADC ne sera active que si l'attribut administrativeState est à l'état déverrouillé.

Le type de commande ADC est identifié par l'attribut adcType qui peut prendre la valeur "centralized" ou "decentralized".

Cet objet de commande n'est instancié que par l'élément de réseau et non par le système d'exploitation lorsqu'il reçoit un message du nœud d'arrivée (type centralisé) ou lorsque le seuil est franchi au niveau de l'élément de réseau de départ (type décentralisé).

Le type et l'indicatif de destination constituent dans cet ordre des clés de la commande.

Lorsque l'objet commande est instancié, seul l'un des attributs d'effet suivants doit être présent: adcPercentage, adcContinuousTimer, adcAsynchronousTimer ou adcLeakyBucket. Cette instance d'objet ADC est supprimée à l'expiration de la temporisation. Si un déclenchement subséquent est reçu avant l'expiration du temporisateur, les valeurs de l'instance ADC sont remplacées par les valeurs correspondant au dernier déclenchement. La valeur de l'attribut autoActivated est toujours Vrai. La classe de cette commande est une sous-classe de la classe d'objets *trafficControl*.";

ATTRIBUTES

adcType GET,
destinationCode GET,
treatment GET,
"CCITT Rec. X.721:1992":administrativeState
DEFAULT VALUE Q823-TM-ASN1Module.defaultAdministrativeState
PERMITTED VALUES Q823-TM-ASN1Module.PermittedState
GET-REPLACE,
autoActivated GET,
timeExpiration GET;

;;

CONDITIONAL PACKAGES

destinationTypePackage

PRESENT IF "si le type de destination est nécessaire pour identifier sans ambiguïté l'indicatif destinationCode",

adcPercentagePackage

PRESENT IF "si les progiciels adcContinuousTimerPackage, adcAsynchronousTimerPackage et adcLeakyBucketPackage sont absents, et si l'élément de réseau le prend en charge.",

adcContinuousTimerPackage

PRESENT IF "si les progiciels adcPercentagePackage, adcAsynchronousTimerPackage et adcLeakyBucketPackage sont absents, et si l'élément de réseau le prend en charge.",

adcAsynchronousTimerPackage

PRESENT IF "si les progiciels adcPercentagePackage, adcContinuousTimerPackage et adcLeakyBucketPackage sont absents, et si l'élément de réseau le prend en charge.",

adcLeakyBucketPackage

PRESENT IF "si les progiciels adcPercentagePackage, adcContinuousTimerPackage et adcAsynchronousTimerPackage sont absents, et si l'élément de réseau le prend en charge.";

REGISTERED AS {q823ObjectClass 4};

11.1.5 adcTrigger

adcTrigger MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721:1992":top;

CHARACTERIZED BY

"ITU-T Rec. M.3100":objectManagementNotificationsPackage,

"ITU-T Rec. M.3100":stateChangeNotificationPackage,

adcTriggerPackage PACKAGE

BEHAVIOUR

adcTriggerBehaviour BEHAVIOUR

DEFINED AS

"L'objet *adcTrigger* est configuré par le système d'exploitation avec les attributs nécessaires de façon à être automatiquement instancié lorsque le seuil de déclenchement est franchi (dans l'élément de réseau où se situe l'objet *adcTrigger* ou dans un élément de réseau adjacent). Le système d'exploitation peut spécifier la destination à surveiller par l'élément de réseau pour connaître l'encombrement, la valeur du seuil de déclenchement, l'effet de la commande et la temporisation.

L'objet *adcTrigger* peut être de type 'centralized', 'decentralized' ou 'both' (centralisé/décentralisé) selon l'attribut *adcType*.

Si cet attribut est de type 'centralized', le mécanisme *adcTrigger* et la commande ADC sont situés dans deux éléments de réseau différents. Un message de signalisation est envoyé de l'élément de réseau de départ (où se situe l'objet *adcTrigger*) vers tous les éléments de réseau adjacents afin qu'ils instancient la commande ADC.

Si l'attribut *adcType* est de type 'decentralized', le mécanisme *adcTrigger* et la commande ADC sont tous deux situés dans l'élément de réseau de départ.

Si l'attribut *adcType* est de type 'both', l'élément de réseau de départ (où se situe le mécanisme *adcTrigger*) envoie un message de signalisation à tous les éléments de réseau adjacents (c'est-à-dire comme pour le type 'centralized') afin qu'ils instancient une commande ADC. Il instancie aussi la commande ADC en lui-même (c'est-à-dire comme pour le type 'decentralized').

Un système d'exploitation peut instancier l'objet *adcTrigger*, mais pas l'objet commande ADC. L'objet *adcTrigger* peut aussi être mis à l'état verrouillé. Quand il est dans cet état, il n'envoie pas automatiquement un message au commutateur adjacent et n'instancie pas la commande ADC même si le seuil de déclenchement est franchi.

Seul l'un des attributs d'effet (*percentage*, *continuousTimer*, *asynchronousTimer* ou *leakyBucket*) doit être présent lorsque l'objet commande est instancié.

Le type et l'indicatif de destination constituent dans cet ordre des clés de la commande.";;

ATTRIBUTES

adcTriggerId GET,

adcTriggerType GET-REPLACE,

"CCITT Rec. X.721:1992":administrativeState

DEFAULT VALUE Q823-TM-ASN1Module.defaultAdministrativeState

PERMITTED VALUES Q823-TM-ASN1Module.PermittedState

GET-REPLACE,

timeExpiration GET-REPLACE;

;;

CONDITIONAL PACKAGES

destinationCodePackage

PRESENT IF "si le déclenchement de la commande ADC est limité à une destination prédéterminée.",

destinationTypePackage

PRESENT IF "si le progiciel destinationCodePackage est présent et si le type de destination est nécessaire pour identifier sans ambiguïté l'indicatif destinationCode.",

triggerThresholdPackage

PRESENT IF "si un seuil doit être défini par le gestionnaire pour activer le mécanisme adcTrigger",

percentagePackage

PRESENT IF "si les progiciels continuousTimerPackage, asynchronousTimerPackage et

leakyBucketPackage sont absents, et si l'élément de réseau le prend en charge.",

continuousTimerPackage

PRESENT IF "si les progiciels percentagePackage, asynchronousTimerPackage et leakyBucketPackage

sont absents, et si l'élément de réseau le prend en charge.",

asynchronousTimerPackage

PRESENT IF "si les progiciels *percentagePackage*, *continuousTimerPackage* et *leakyBucketPackage* sont absents, et si l'élément de réseau le prend en charge.",
leakyBucketPackage
 PRESENT IF "si les progiciels *percentagePackage*, *continuousTimerPackage* et *asynchronousTimerPackage* ont absents, et si l'élément de réseau le prend en charge.";
 REGISTERED AS {q823ObjectClass 5};

11.1.6 cancelFrom

cancelFrom MANAGED OBJECT CLASS
 DERIVED FROM trafficControl;
 CHARACTERIZED BY
 cancelFromPackage PACKAGE
 BEHAVIOUR
 cancelFromBehaviour BEHAVIOUR
 DEFINED AS
 "Cette commande est activée manuellement sur un sous-faisceau de circuits sortant et empêche le trafic de déborder sur les sous-faisceaux suivants de la chaîne. Elle est définie au 3.2.1/E.412.
 Cette commande est protectrice et s'applique aux sous-faisceaux de circuits. Sa classe est une sous-classe de la classe *trafficControl*.
 Elle peut servir à empêcher le débordement des trafics de premier choix et de débordement. La distinction entre ces deux types de trafic est faite par l'affectation d'une valeur de type appropriée à l'attribut *routingAspect*.
 L'attribut *controlTmCircuitEndPointSubgroup* constitue la clé de priorité la plus élevée, suivi, dans l'ordre, par chaque composante des attributs *originationAspect*, *destinationAspect* et *routingAspect*.
 La valeur la plus spécifique est prioritaire.";;
 ATTRIBUTES
 controlTmCircuitEndPointSubgroup GET,
 routingAspect REPLACE-WITH-DEFAULT
 DEFAULT VALUE Q823-TM-ASN1Module.defaultRoutingAspects
 GET-REPLACE,
 destinationAspect REPLACE-WITH-DEFAULT
 DEFAULT VALUE Q823-TM-ASN1Module.defaultDestinationAspect
 GET-REPLACE,
 originationAspect REPLACE-WITH-DEFAULT
 DEFAULT VALUE Q823-TM-ASN1Module.defaultOriginAspect
 GET-REPLACE,
 percentage GET-REPLACE,
 treatment GET-REPLACE;
 ;;
 REGISTERED AS {q823ObjectClass 6};

11.1.7 cancelRerouted

cancelRerouted MANAGED OBJECT CLASS
 DERIVED FROM trafficControl;
 CHARACTERIZED BY
 cancelReroutedPackage PACKAGE
 BEHAVIOUR
 cancelReroutedBehaviour BEHAVIOUR
 DEFINED AS
 "Cette commande empêche le réacheminement ou le débordement du trafic qui a déjà été réacheminé. Le trafic réacheminé n'est pas autorisé à déborder sur le sous-faisceau auquel s'applique la commande *cancelRerouted*, tandis que le trafic de débordement normal n'en est pas affecté. Cette commande est définie au 3.2.4/E.412.
 Il est à noter que pour appliquer cette commande, chaque appel doit être marqué par un message de signalisation chaque fois qu'une commande de débordement temporaire TAR lui est appliquée.
 Cette commande est manuelle et protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*.";;

ATTRIBUTES
controlTmCircuitEndPointSubgroup GET,
treatment GET-REPLACE
;;
;
REGISTERED AS {q823ObjectClass 7};

11.1.8 cancelTo

cancelTo MANAGED OBJECT CLASS
DERIVED FROM trafficControl;
CHARACTERIZED BY

cancelToPackage PACKAGE

BEHAVIOUR

cancelToBehaviour BEHAVIOUR

DEFINED AS

"Cette commande concerne l'annulation des trafics de premier choix et de débordement vers un sous-faisceau *controlTmCircuitEndPointSubgroup* donné.

La commande d'annulation du trafic de premier choix empêche ce trafic d'accéder à un sous-faisceau de circuits. Elle est définie au 3.1.2/E.412.

La commande d'annulation du trafic de débordement est activée sur un sous-faisceau sortant et empêche ce trafic d'accéder au sous-faisceau auquel la commande s'applique. Elle est définie au 3.2.1/E.412.

La distinction entre ces deux types de trafic se fait en attribuant à l'attribut *routingAspect* une valeur appropriée.

L'Appendice IV explique comment empêcher les trafics de premier choix et de débordement d'accéder à un sous-faisceau en utilisant la commande *cancelTo*.

L'attribut *controlTmCircuitEndPointSubgroup* constitue la clé de priorité la plus élevée, suivi par chacune des composantes des attributs *originationAspect*, *destinationAspect* et *routingAspect*. La valeur la plus spécifique est prioritaire.

Cette commande est manuelle et protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*.";;

ATTRIBUTES

controlTmCircuitEndPointSubgroup	GET,
routingAspect	REPLACE-WITH-DEFAULT DEFAULT VALUE Q823-TM-ASN1Module.defaultRoutingAspects GET-REPLACE,
destinationAspect	REPLACE-WITH-DEFAULT DEFAULT VALUE Q823-TM-ASN1Module.defaultDestinationAspect GET-REPLACE,
originationAspect	REPLACE-WITH-DEFAULT DEFAULT VALUE Q823-TM-ASN1Module.defaultOriginAspect GET-REPLACE,
percentage	GET-REPLACE,
treatment	GET-REPLACE;
;;	

REGISTERED AS {q823ObjectClass 8};

11.1.9 CircuitEndPointSubgroupCurrentData

circuitEndPointSubgroupCurrentData MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Rec. Q.822:1994":currentData;
CHARACTERIZED BY

circuitEndPointSubgroupCurrentDataPackage PACKAGE

BEHAVIOUR

circuitEndPointSubgroupCurrentDataBehaviour BEHAVIOUR

DEFINED AS

"La classe *circuitEndPointSubgroupCurrentData* est une sous-classe de la classe *currentData*. Elle sert à surveiller les données de performance des sous-faisceaux de circuits conformément à la définition de la Recommandation E.502.

Les attributs de surveillance des performances des sous-faisceaux de circuits dépendent de la directionnalité des sous-faisceaux; ces derniers peuvent être entrants, sortants ou bidirectionnels.

A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou toute autre classe qui en dérive.";;

ATTRIBUTES

incomingSeizures
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
outgoingBids
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
outgoingSeizures
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
answeredOutgoingSeizures
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
overflow
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
incomingTrafficUsage
INITIAL VALUE Q823-TM-ASN1Module.initialInteger,
outgoingTrafficUsage
INITIAL VALUE Q823-TM-ASN1Module.initialInteger,
numberOfAvailCircuits
INITIAL VALUE Q823-TM-ASN1Module.initialGauge;
;;

CONDITIONAL PACKAGES

answeredIncomingSeizuresPackage

PRESENT IF "si cette mesure de performance est prise en charge par l'élément de réseau";

REGISTERED AS {q823ObjectClass 9};

11.1.10 circuitEndPointSubgroupHistoryData

circuitEndPointSubgroupHistoryData MANAGED OBJECT CLASS

DERIVED FROM "ITU-T Rec. Q.822:1994":historyData;

CHARACTERIZED BY

circuitEndPointSubgroupHistoryDataPackage PACKAGE

BEHAVIOUR

circuitEndPointSubgroupHistoryDataBehaviour BEHAVIOUR

DEFINED AS

"La classe *circuitEndPointSubgroupHistoryData* dans le cadre de la gestion de trafic est une sous-classe de la classe d'objets *historyData*. Elle sert à surveiller les données de performance de sous-faisceau de circuits conformément à la définition de la Recommandation E.502.

Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante.";;

ATTRIBUTES

incomingSeizures GET,
outgoingBids GET,
outgoingSeizures GET,
answeredOutgoingSeizures GET,
overflow GET,
incomingTrafficUsage GET,
outgoingTrafficUsage GET,
numberOfAvailCircuits GET;
;;

CONDITIONAL PACKAGES

answeredIncomingSeizureHistoryPackage

PRESENT IF "si ce progiciel est présent dans la classe *currentData* correspondante";

REGISTERED AS {q823ObjectClass 10};

11.1.11 congestionLevelIndication

congestionLevelIndication MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721:1992":top;

CHARACTERIZED BY

"ITU-T Rec. M.3100":attributeValueChangeNotificationPackage,
congestionLevelIndicationPackage PACKAGE

BEHAVIOUR

congestionLevelIndicationBehaviour BEHAVIOUR

DEFINED AS

"Une instance de la classe *congestionLevelIndication* fournit une indication du niveau d'encombrement courant de l'instance d'objet d'élément géré dans laquelle l'instance de cette classe est implantée. Le système d'exploitation n'est pas autorisé à instancier cette classe.";;

ATTRIBUTES

congestionLevelIndicationId GET,
congestionLevel GET;

;;

REGISTERED AS {q823ObjectClass 11};

11.1.12 destinationCodeControl

destinationCodeControl MANAGED OBJECT CLASS

DERIVED FROM trafficControl;

CHARACTERIZED BY

destinationCodeControlPackage PACKAGE

BEHAVIOUR

destinationCodeControlBehaviour BEHAVIOUR

DEFINED AS

"Cette commande manuelle interdit l'acheminement vers une destination donnée selon un pourcentage (se rapporter au paragraphe intitulé blocage sur indicatif de la Recommandation E.412) ou un nombre d'appels (se rapporter au paragraphe intitulé espacement d'appels de la Recommandation E.412).

Elle peut s'appliquer à un indicatif de pays ou de zone ou à un code d'identification de commutateur ou au numéro d'un autre lieu. Des précisions sont données au 3.1.1.1/E.412.

Lorsque cette commande est active en mode blocage sur indicatif, le pourcentage spécifié d'appels est bloqué (annulation).

Lorsqu'elle est active en mode espacement d'appels, le nombre d'appels libérés pour cette destination est commandé. Il peut être spécifié par l'un des trois attributs suivants: *continuousTimer*, *asynchronousTimer* ou *leakyBucket*.

L'effet de cette commande (blocage sur indicatif ou espacement d'appels) peut être défini à l'intérieur de la même classe ou dans une classe séparée. Le mode blocage sur indicatif s'applique lorsque l'effet ne concerne qu'une seule destination. Dans ce cas, il existe quatre attributs (*percentage*, *continuousTimer*, *asynchronousTimer* ou *leakyBucket*) qui spécifient cet effet dans la même classe; seul l'un d'entre eux doit être spécifié. Dans ce scénario, l'instance d'un seul objet agit comme une fonction de commande d'indicatif. Le mode espacement d'appels définit l'effet de cette commande dans une classe séparée lorsque celui-ci s'applique à tout un groupe de destinations (le compteur à fuite montre où l'utiliser). Dans ce cas, il est nécessaire que l'attribut *assocOwnerDccGroup* pointe vers la classe contenant l'effet de la commande. Dans ce scénario, deux classes d'objets représentent ensemble une fonction de commande d'indicatif.

Les règles suivantes sont valides indépendamment du nombre de classes utilisé pour représenter la fonction de commande d'indicatif:

- il est impossible de créer deux ou plusieurs instances d'objet *destinationCodeControl* avec les mêmes combinaisons de valeurs d'attributs clés: *destinationType*, *destinationCode* et *originationAspect*;

- l'attribut *destinationCode* constitue la clé de priorité la plus élevée, suivi dans cet ordre de l'attribut *destinationType* (s'il est présent) et des composantes de l'attribut *originationAspect*.

Lorsqu'il existe plus d'une commande avec des numéros qui se chevauchent, la commande la plus spécifique du même type s'applique à la place de la commande la moins spécifique, chacune d'elles pouvant affecter le même appel.

Si l'attribut *assocOwnerDccGroup* est spécifié, alors l'attribut *dccgroup* associé doit exister. Les instances de cette classe appartiennent au même groupe si et seulement si elles ont la même valeur d'attribut *assocOwnerDccGroup*.

La classe de cette commande est une sous-classe de la classe d'objets *trafficControl*.";;

ATTRIBUTES

destinationCode GET,
originationAspect REPLACE-WITH-DEFAULT
DEFAULT VALUE Q823-TM-ASN1Module.defaultOriginAspect
GET-REPLACE,
treatment GET-REPLACE;

;;

CONDITIONAL PACKAGES

destinationTypePackage

PRESENT IF "si le type de destination est nécessaire pour identifier sans ambiguïté l'indicatif destinationCode.",

percentagePackage

PRESENT IF "si les progiciels *continuousTimerPackage*, *asynchronousTimerPackage*, *leakyBucketPackage* et *assocOwnerDccGroupPackage* sont absents, et si l'élément de réseau le prend en charge.",

continuousTimerPackage

PRESENT IF "si les progiciels *percentagePackage*, *asynchronousTimerPackage*, *leakyBucketPackage* et *assocOwnerDccGroupPackage* sont absents, et si l'élément de réseau le prend en charge.",

asynchronousTimerPackage

PRESENT IF "si les progiciels *percentagePackage*, *continuousTimerPackage*, *leakyBucketPackage* et *assocOwnerDccGroupPackage* sont absents, et si l'élément de réseau le prend en charge.",

leakyBucketPackage

PRESENT IF "si les progiciels *percentagePackage*, *continuousTimerPackage*, *asynchronousTimerPackage* et *assocOwnerDccGroupPackage* sont absents, et si l'élément de réseau le prend en charge.",

assocOwnerDccGroupPackage

PRESENT IF "si les progiciels *percentagePackage*, *continuousTimerPackage*, *asynchronousTimerPackage* et *leakyBucketPackage* sont absents, et si l'élément de réseau le prend en charge.";

REGISTERED AS {q823ObjectClass 12};

11.1.13 dccGroup

dccGroup MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721:1992":top;

CHARACTERIZED BY

"ITU-T Rec. M.3100":objectManagementNotificationsPackage,
dccGroupPackage PACKAGE

BEHAVIOUR

dccGroupBehaviour BEHAVIOUR

DEFINED AS

"Cette classe définit l'effet de la commande d'indicatif de destination lorsqu'on souhaite appliquer le même effet à tout un groupe de destinations. Ses instances sont associées aux instances de la classe *destinationCodeControl* par une relation (1:n). Cette classe ne fournit pas par elle-même de fonction de commande d'indicatif de destination; quand une ou plusieurs de ses instances sont associées à une ou plusieurs des instances *destinationCodeControl*, elle représente la fonction de commande d'indicatif de destination.

Seul l'un des attributs d'effet (*percentage*, *continuousTimer*, *asynchronousTimer* ou *leakyBucket*) doit être présent lors de l'instance de l'objet commande.";

ATTRIBUTES

dccGroupId GET;

;;

CONDITIONAL PACKAGES

percentagePackage

PRESENT IF "si les progiciels *continuousTimerPackage*, *asynchronousTimerPackage* et *leakyBucketPackage* sont absents, et si l'élément de réseau le prend en charge.",

continuousTimerPackage

PRESENT IF "si les progiciels *percentagePackage*, *asynchronousTimerPackage* et *leakyBucketPackage* sont absents, et si l'élément de réseau le prend en charge.",

asynchronousTimerPackage

PRESENT IF "si les progiciels *percentagePackage*, *continuousTimerPackage* et *leakyBucketPackage* sont absents, et si l'élément de réseau le prend en charge.",

leakyBucketPackage

PRESENT IF "si les progiciels *percentagePackage*, *continuousTimerPackage* et *asynchronousTimerPackage* sont absents, et si l'élément de réseau le prend en charge.";

REGISTERED AS {q823ObjectClass 13};

11.1.14 exchangeCurrentData

exchangeCurrentData MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Rec. Q.822:1994":currentData;
CHARACTERIZED BY
 exchangeCurrentDataPackage PACKAGE
BEHAVIOUR
 exchangeCurrentDataBehaviour BEHAVIOUR
DEFINED AS

"La classe *exchangeCurrentData* est une sous-classe de la classe d'objets *CurrentData*. Elle sert à surveiller les données de performance du commutateur conformément à la définition de la Recommandation E.502. A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou toute autre classe qui en dérive.
Les attributs de surveillance des performances du commutateur dépendent des caractéristiques de flux des principaux types de trafic montrés sur la Figure 4/E.502";;

ATTRIBUTES

incomingTraffic
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
outgoingTraffic
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
transitTraffic
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
terminatingTraffic
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
originatingTraffic
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
internalTraffic
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
callsBlockedByLoadShedding
INITIAL VALUE Q823-TM-ASN1Module.initialCount;
;;

REGISTERED AS {q823ObjectClass 14};

11.1.15 exchangeHistoryData

exchangeHistoryData MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Rec. Q.822:1994":historyData;
CHARACTERIZED BY
 exchangeHistoryDataPackage PACKAGE
BEHAVIOUR
 exchangeHistoryDataBehaviour BEHAVIOUR
DEFINED AS

"La classe *exchangeHistoryData* est une sous-classe de la classe *historyData*. Elle sert à surveiller les données de performance de commutateur conformément à la définition de la Recommandation E.502. Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante.
Les attributs de surveillance des performances d'un commutateur dépendent des caractéristiques de flux des principaux types de trafic montrés sur la Figure 4/E.502.";

ATTRIBUTES

incomingTraffic GET,
outgoingTraffic GET,
transitTraffic GET,
terminatingTraffic GET,
originatingTraffic GET,
internalTraffic GET,
callsBlockedByLoadShedding GET
;;

;
REGISTERED AS {q823ObjectClass 15};

11.1.16 htrDestination

htrDestination MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721:1992":top;
CHARACTERIZED BY

"ITU-T Rec. M.3100":objectManagementNotificationsPackage,
"ITU-T Rec. M.3100":stateChangeNotificationPackage,
creatorPackage,
htrDestinationPackage PACKAGE

BEHAVIOUR

htrDestinationBehaviour BEHAVIOUR

DEFINED AS

"Une instance de l'objet *htrDestination* représente une destination qui est identifiée comme difficile d'accès. Il appartient au système d'exploitation ou de service de gestion des ressources du commutateur de décider si une destination est facile ou difficile d'accès; cette décision est prise suite à une information extérieure (par exemple tremblement de terre) ou en fonction du taux de tentatives de prise avec réponse ou du taux de prises avec réponse. L'attribut *administrativeState* permet de verrouiller l'état difficile d'accès de façon que temporairement il n'en soit tenu aucun compte. L'état HTR d'une destination peut aussi être corrélé avec les sous-faisceaux de circuits. Si l'attribut *administrativeState* est vide alors la destination est considérée comme HTR pour tous les sous-faisceaux de circuits possibles.

Une telle instance peut être créée explicitement par le gestionnaire (en cas de destination HTR établie en mode manuel) ou automatiquement par l'élément de réseau (en cas de détermination automatique par l'agent).

Dans ce modèle, le mécanisme de reconnaissance de l'état difficile ou facile d'accès par le système de gestion des ressources du commutateur est local: il n'est donc pas modélisé.

Une destination qui est inhibée (*administrativeState* = verrouillé) ou pour laquelle aucun objet *htrDestination* n'est instancié doit être considérée comme facile d'accès.

Toutes les instances de la classe d'objets *htrDestination* forment la liste HTR.";;

ATTRIBUTES

htrDestinationId GET,
destinationCode GET,
"CCITT Rec. X.721:1992":administrativeState
DEFAULT VALUE Q823-TM-ASN1Module.defaultAdministrativeState
PERMITTED VALUES Q823-TM-ASN1Module.PermittedState
GET-REPLACE;

;;

CONDITIONAL PACKAGES

destinationTypePackage
PRESENT IF "si le type de destination est nécessaire pour identifier sans ambiguïté l'indicatif de destination",
tmCircuitEndPointSubgroupListPackage
PRESENT IF "s'il y a corrélation entre la destination *htrDestination* et les sous-faisceaux *tmCircuitEndPointSubgroups*";

REGISTERED AS {q823ObjectClass 16};

11.1.17 observedDestination

observedDestination MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721:1992":top;
CHARACTERIZED BY

"ITU-T Rec. M.3100":objectManagementNotificationsPackage,
observedDestinationPackage PACKAGE

BEHAVIOUR

observedDestinationBehaviour BEHAVIOUR

DEFINED AS

L'objet *observedDestination* est instancié quand une destination doit être surveillée à des fins de gestion des performances, par exemple pour déterminer si elle est difficile d'accès. La destination est un pays, une zone, un commutateur ou un autre emplacement dans lequel le demandé est situé; elle peut être spécifiée à l'intérieur du pays. Elle est définie à l'Annexe A/E.410. Une destination peut aussi être observée de façon sélective sur un ensemble de sous-faisceaux de circuits. Si l'objet *tmCircuitEndPointSubgroupList* n'est pas présent, la destination sera observée pour tous les sous-faisceaux de circuits.";;

ATTRIBUTES

observedDestinationId GET,
destinationCode GET,
tmSurveillance GET-REPLACE;
;;

CONDITIONAL PACKAGES

destinationTypePackage
PRESENT IF "si le type de destination est nécessaire pour identifier sans ambiguïté l'indicatif de destination",
creatorPackage
PRESENT IF "si une instance le prend en charge",
tmCircuitEndPointSubgroupListPackage
PRESENT IF "les performances de la destination doivent être surveillées en corrélation avec certains sous-faisceaux *tmCircuitEndPointSubgroups*";

REGISTERED AS {q823ObjectClass 17};

11.1.18 observedDestinationCurrentData

observedDestinationCurrentData MANAGED OBJECT CLASS

DERIVED FROM "ITU-T Rec. Q.822:1994":currentData;

CHARACTERIZED BY

observedDestinationCurrentDataPackage PACKAGE

BEHAVIOUR

observedDestinationCurrentDataBehaviour BEHAVIOUR

DEFINED AS

"La classe *observedDestinationCurrentData* est une sous-classe de la classe d'objets *CurrentData*. Elle sert à surveiller les performances de destination conformément à la définition de la Recommandation E.502.

A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou une sous-classe qui en dérive.";;

ATTRIBUTES

bids
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
outgoingSeizures
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
answeredOutgoingSeizures
INITIAL VALUE Q823-TM-ASN1Module.initialCount,
noCircuitsAvailable
INITIAL VALUE Q823-TM-ASN1Module.initialCount;
;;

CONDITIONAL PACKAGES

callsAffectedByDccPackage PRESENT IF "si une instance le prend en charge ";

REGISTERED AS {q823ObjectClass 18};

11.1.19 observedDestinationHistoryData

observedDestinationHistoryData MANAGED OBJECT CLASS

DERIVED FROM "ITU-T Rec. Q.822:1994":historyData;

CHARACTERIZED BY

observedDestinationHistoryDataPackage PACKAGE

BEHAVIOUR

observedDestinationHistoryDataBehaviour BEHAVIOUR

DEFINED AS

"La classe *observedDestinationHistoryData* est une sous-classe de la classe *historyData*. Elle sert à surveiller les données de performance de destination conformément à la définition de la Recommandation E.502. Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante.";;

ATTRIBUTES

bids GET,
outgoingSeizures GET,
answeredOutgoingSeizures GET,
noCircuitsAvailable GET;
;;

CONDITIONAL PACKAGES

callsAffectedByDccHistoryPackage

PRESENT IF "ce progiciel est présent s'il est présent dans l'instance *currentData* correspondante";

REGISTERED AS {q823ObjectClass 19};

11.1.20 scr

scr MANAGED OBJECT CLASS

DERIVED FROM trafficControl;

CHARACTERIZED BY

"ITU-T Rec. M.3100":stateChangeNotificationPackage,
scrPackage PACKAGE

BEHAVIOUR

scrBehaviour BEHAVIOUR

DEFINED AS

"La commande SCR permet à un commutateur de donner automatiquement la préférence à certains attributs par rapport à d'autres (le trafic de premier choix sur le trafic de débordement, par exemple) lorsque les circuits sont encombrés ou sur le point de l'être. Cette commande peut être fournie avec un ou deux seuils et devenir ultérieurement plus sélective. Elle est préassignée avec des seuils fixant le nombre de circuits à laisser au repos selon les différents types de trafic offerts à un sous-faisceau. Si ce nombre de circuits au repos ou la capacité de réserve du sous-faisceau est inférieure ou égale au seuil de réservation, l'appel est annulé ou dévié vers le sous-faisceau suivant de la chaîne. Cette commande est définie au 4.2/E.412.

La commande SCR se compose des paramètres suivants:

seuils de réservation

réponse à une commande
traitement.

Les seuils de réservation et les réponses des commandes, qui leur sont associées, sont déterminés par l'attribut *activationThresholds* et l'instance d'objet *scrAffectedTraffic* associée. Le traitement des appels, dont l'accès au sous-faisceau a été refusé, est donné dans l'attribut *dispositionOfCalls*.

Si ce nombre de circuits au repos ou la capacité de réserve du sous-faisceau est inférieure ou égale au seuil de réservation, le commutateur vérifiera l'instance d'objet *scrAffectedTraffic* indiquée, afin de déterminer si l'appel sera commandé. Les types de traitements pouvant servir à étrangler le trafic sont annulation et évitement.

Les niveaux définis dans l'attribut *activationThresholds* et la référence donnée aux attributs de trafic réservé SCR composite correspondront aux commandes à seuil unique ou multiple.

L'instance d'objet *scrAffectedTraffic* doit être présente avant l'instance de la classe SCR.

Cette commande est automatique et protectrice. Sa classe est une sous-classe de la classe d'objets *trafficControl*.";;

ATTRIBUTES

controlTmCircuitEndPointSubgroup GET,
dispositionOfCalls GET-REPLACE,
"CCITT Rec. X.721:1992":administrativeState

```

                                DEFAULT      VALUE      Q823-TM-ASN1Module
defaultAdministrativeState
                                PERMITTED VALUES Q823-TM-ASN1Module.PermittedState
                                GET-REPLACE,
                                autoActivated      GET,
                                activationThresholds  GET-REPLACE,
                                assocScrAffectedTraffic GET-REPLACE;
                                ;;
REGISTERED AS {q823ObjectClass 20};

```

11.1.21 scrAffectedTraffic

```

scrAffectedTraffic MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721:1992":top;
CHARACTERIZED BY
    "ITU-T Rec. M.3100":objectManagementNotificationsPackage,
    scrAffectedTrafficPackage PACKAGE
BEHAVIOUR
    scrAffectedTrafficBehaviour BEHAVIOUR
DEFINED AS
    "La classe d'objets scrAffectedTraffic représente la catégorie de réponse des commandes SCR.
    Elle détermine l'effet de cette commande sur le trafic pour chacun des aspects suivants: acheminement,
    destination, origine ou autres critères.
    Si l'attribut level2ResponseCategories est spécifié, les attributs activationThreshold de niveaux 1 et 2 doivent
    être spécifiés avec la même unité (nombre ou pourcentage).
    Les aspects liés à l'origine, à la destination et à l'acheminement et à d'autres critères constituent dans cet
    ordre des clés.";;
ATTRIBUTES
    scrAffectedTrafficId      GET,
    level1ResponseCategories  GET-REPLACE;
    ;;
CONDITIONAL PACKAGES
    level2ResponseCategoriesPackage
    PRESENT IF "l'instance prend en charge la réservation sélective de circuit multiseuil";
REGISTERED AS {q823ObjectClass 21};

```

11.1.22 skip

```

skip MANAGED OBJECT CLASS
DERIVED FROM trafficControl;
CHARACTERIZED BY
    skipPackage PACKAGE
BEHAVIOUR
    skipBehaviour BEHAVIOUR
DEFINED AS
    "Cette commande est activée sur un faisceau de circuits sortant et sert à obliger le trafic à emprunter le
    prochain sous-faisceau de la table d'acheminement. Elle peut affecter les trafics de premier choix et de
    débordement. Elle est définie au 3.2.2/E.412.
    L'attribut controlTmCircuitEndPointSubgroup constitue la clé de priorité la plus élevée, suivi de chaque
    composante des attributs originationAspect, destinationAspect et routingAspect dans cet ordre. La valeur la
    plus spécifique a la priorité.
    Cette commande est manuelle et protectrice. Sa classe est une sous-classe de la classe d'objets
    trafficControl.";;
ATTRIBUTES
    controlTmCircuitEndPointSubgroup  GET,
    routingAspect                      REPLACE-WITH-DEFAULT
                                DEFAULT VALUE Q823-TM-ASN1Module.defaultRoutingAspects
    destinationAspect                 REPLACE-WITH-DEFAULT
                                DEFAULT VALUE Q823-TM-ASN1Module.defaultDestinationAspect
                                GET-REPLACE,

```

```

    originationAspect          REPLACE-WITH-DEFAULT
                                DEFAULT VALUE Q823-TM-ASN1Module.defaultOriginAspect
    percentage                  GET-REPLACE,
                                GET-REPLACE
    ;;
;
REGISTERED AS {q823ObjectClass 22};

```

11.1.23 StateIndicator

```

stateIndicator MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Rec. X.739:1993":scanner;
CHARACTERIZED BY
stateIndicatorPackage PACKAGE
BEHAVIOUR
stateIndicatorBehaviour BEHAVIOUR
DEFINED AS

```

"Cette classe définit les actions et les notifications contenant un bit pour chaque condition. Si à tout moment lors de la période, la condition a été satisfaite, la valeur du bit est égale à 1; sinon sa valeur est égale à 0. On définit les bits suivants:

- a) *exchangeCongestionLevel1* – 1^{er} niveau d'encombrement du commutateur (CL1): ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau se trouve au 1^{er} niveau d'encombrement.
- b) *exchangeCongestionLevel2* – 2^{ème} niveau d'encombrement du commutateur (CL2): ce bit est égal à 1 si durant la période *granularityPeriod*, l'élément de réseau se trouve au 2^{ème} niveau d'encombrement.
- c) *congestionLevel1Received* – indicateur CL1 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL1 d'un des commutateurs adjacents.
- d) *congestionLevel2Received* – indicateur CL2 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL2 d'un des commutateurs adjacents.
- e) *scrTriggered* – déclenchement SCR: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur 'unlocked' et si l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande SCR.
- f) *AccTriggered* – déclenchement ACC: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur 'unlocked' et si l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande ACC.
- g) *protectiveControlActive* – commande protectrice activée: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes manuelles protectrices (c'est-à-dire: *cancelTo*, *cancelFrom*, *skip*, *cancelRerouted*) sont présentes dans l'élément de réseau.
- h) *expansiveControlActive* – commande expansive activée: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes manuelles expansives (c'est-à-dire: *tarTo*, *tarFrom*) sont présentes dans l'élément de réseau.
- i) *destinationControlActive* – activation de commande de destination: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes manuelles de destination (c'est-à-dire: fonctions *destinationCodeControl*) sont présentes dans l'élément de réseau.
- j) *htrDestinationActive* – activation de commande de destination: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes *htrDestination* sont présentes dans l'élément de réseau.
- k) *circuitEndPointSubgroupAddedOrDeleted* – addition ou suppression d'un sous-faisceau d'extrémité de circuits: ce bit est égal à 1 durant la période *granularityPeriod* si un sous-faisceau est ajouté ou supprimé dans l'élément de réseau.
- l) *accTransmissionInhibited* – transmission acc inhibée: ce bit est égal à 1 si durant la période *granularityPeriod*, l'attribut *administrativeState* du mécanisme *accTrigger* est à l'état verrouillé.
- m) *adcTriggered* – commande adc déclenchée: ce bit est égal à 1 si durant la période *granularityPeriod*, l'attribut *administrativeState* est à l'état déverrouillé et si l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande ADC.

La période *granularityPeriod* de cette classe sera inférieure ou égale à celle utilisée pour la classe *currentData* définie dans la présente Recommandation; elle sera spécifiée en secondes ou en minutes.";;;;

ATTRIBUTES

stateIndicatorId GET;

CONDITIONAL PACKAGES

stateIndicatorActionPackage

PRESENT IF "le progiciel stateIndicatorNotificationPackage est absent",

stateIndicatorNotificationPackage

PRESENT IF "le progiciel stateIndicatorActionPackage est absent ";

REGISTERED AS {q823ObjectClass 23};

11.1.24 tarFrom

tarFrom MANAGED OBJECT CLASS

DERIVED FROM trafficControl;

CHARACTERIZED BY

tarFromPackage PACKAGE

BEHAVIOUR

tarFromBehaviour BEHAVIOUR

DEFINED AS

"La commande *tarFrom* est une commande expansive qui augmente temporairement le nombre de possibilités d'acheminement afin d'établir des appels par:

- addition de nouveaux sous-faisceaux à la fin de la table d'acheminement,
- insertion de nouveaux faisceaux entre les sous-faisceaux déjà existants de cette table afin de fournir des voies de débordement supplémentaires, qui sont normalement indisponibles dans la table.

Ces deux types de débordements temporaires (addition à la suite ou insertion entre des sous-faisceaux) sont décrits au 3.2.3/E.412. Le positionnement sur évitement de l'attribut *returnAction* de la commande *tarFrom* n'est pas valide. Cette commande ne s'applique qu'au niveau des sous-faisceaux et agit sur toutes les tables d'acheminement où l'attribut *controlTmCircuitEndPointSubgroup* apparaît. Les faisceaux de circuits de débordement temporaire doivent se terminer sur un commutateur qui a la capacité d'atteindre la destination finale.

L'effet de cette commande peut aussi être limité aux destinations HTR en positionnant l'attribut *destinationAspect* ou en spécifiant explicitement la destination avec l'attribut *destinationCode*. Si l'indicatif de destination est une chaîne vide, la commande s'applique aux destinations que ce sous-faisceau dessert. Il est aussi possible de limiter l'effet de cette commande au trafic de premier choix ou de débordement. Les attributs *controlTmCircuitEndPointSubgroup*, *destinationType* (s'il est présent), *destinationCode*, les composantes des attributs *originationAspect*, *destinationAspect* et *routingAspect* constituent dans cet ordre des clés de la commande.

Cette commande est manuelle et expansive. Sa classe est une sous-classe de la classe d'objets *trafficControl*.";

ATTRIBUTES

controlTmCircuitEndPointSubgroup GET,

newTmCircuitEndPointSubgroups GET-REPLACE,

routingAspect REPLACE-WITH-DEFAULT

DEFAULT VALUE Q823-TM-ASN1Module.defaultRoutingAspects

GET-REPLACE,

destinationAspect REPLACE-WITH-DEFAULT

DEFAULT VALUE Q823-TM-ASN1Module.defaultDestinationAspect

GET-REPLACE,

originationAspect REPLACE-WITH-DEFAULT

DEFAULT VALUE Q823-TM-ASN1Module.defaultOriginAspect

GET-REPLACE,

percentage GET-REPLACE,

destinationCode DEFAULT VALUE Q823-TM-ASN1Module.defaultAllDestinationCodes

GET,

returnAction PERMITTED VALUES Q823-TM-ASN1Module.PermittedTarFromReturnAction

GET

;;

;

CONDITIONAL PACKAGES

destinationTypePackage

PRESENT IF "si le type de destination est nécessaire pour identifier sans ambiguïté l'indicatif

destinationCode";

REGISTERED AS {q823ObjectClass 24};

11.1.25 tarTo

tarTo MANAGED OBJECT CLASS

DERIVED FROM trafficControl;

CHARACTERIZED BY

tarToPackage PACKAGE

BEHAVIOUR

tarToBehaviour BEHAVIOUR

DEFINED AS

"La commande *tarTo* est une commande expansive qui augmente temporairement le nombre de possibilités d'acheminement pour établir les appels soit en ajoutant de nouveaux sous-faisceaux au début de la table d'acheminement de façon que le trafic soit d'abord offert à ces nouveaux sous-faisceaux, soit en remplaçant les sous-faisceaux existants par un ensemble de nouveaux sous-faisceaux de circuits. Un ou plusieurs faisceaux de circuits, qui sont normalement indisponibles dans la table d'acheminement normale mais qui sont au repos, sont rendus disponibles.

Ces deux types de débordement (addition avant ou remplacement de sous-faisceaux existants) sont décrits au 3.2.3/E.412. Le positionnement de l'attribut *returnAction* sur la valeur Néant provoque l'insertion de sous-faisceaux *newTmCircuitEndPointSubgroups* avant le sous-faisceau *controlTmCircuitEndPointSubgroup*. Son positionnement sur évitement provoque le remplacement du sous-faisceau *controlTmCircuitEndPointSubgroup* par les sous-faisceaux *newTmCircuitEndPointSubgroups*. Son positionnement sur traitement provoque le remplacement du sous-faisceau *controlTmCircuitEndPointSubgroup* et des sous-faisceaux de circuits restants du plan d'acheminement par les sous-faisceaux *newTmCircuitEndPointSubgroups*.

La commande *tarTo* ne s'applique qu'au niveau des sous-faisceaux et agit sur toutes les tables d'acheminement où le sous-faisceau *TmCircuitEndPointSubgroup* est présent. Les faisceaux de circuits de débordement temporaire doivent se terminer sur un commutateur qui a la capacité d'atteindre la destination finale.

L'effet de cette commande peut se limiter aux destinations de la liste *HTR* en positionnant l'attribut *routingAspect* ou en spécifiant explicitement la destination dans l'attribut *destinationCode*. Si ce dernier est une chaîne vide, la commande est valide pour toutes les destinations du sous-faisceau *TmCircuitEndPointSubgroup*.

Il est aussi possible de limiter l'effet de cette commande au trafic de premier choix ou de débordement. Les attributs *controlTmCircuitEndPointSubgroup*, *destinationType* (s'il est présent), *destinationCode*, les composantes des attributs *originationAspect*, *destinationAspect* et *routingAspect* constituent dans cet ordre des clés de la commande.

Cette commande est manuelle et expansive. Sa classe est une sous-classe de la classe d'objets *trafficControl*.";

ATTRIBUTES

controlTmCircuitEndPointSubgroup	GET,
newTmCircuitEndPointSubgroups	GET-REPLACE,
routingAspect	REPLACE-WITH-DEFAULT
	DEFAULT VALUE Q823-TM-ASN1Module.defaultRoutingAspects
	GET-REPLACE,
destinationAspect	REPLACE-WITH-DEFAULT
	DEFAULT VALUE Q823-TM-ASN1Module.defaultDestinationAspect
	GET-REPLACE,
originationAspect	REPLACE-WITH-DEFAULT
	DEFAULT VALUE Q823-TM-ASN1Module.defaultOriginAspect
	GET-REPLACE,
percentage	GET-REPLACE,
destinationCode	DEFAULT VALUE Q823-TM-ASN1Module.defaultAllDestinationCodes
	GET
returnAction	GET;
	;;

CONDITIONAL PACKAGES

destinationTypePackage

PRESENT IF "si le type de destination est nécessaire pour identifier sans ambiguïté l'indicatif
destinationCode";

REGISTERED AS {q823ObjectClass 25};

11.1.26 tmCircuitEndPointSubgroup

tmCircuitEndPointSubgroup MANAGED OBJECT CLASS

DERIVED FROM "ITU-T Rec. M.3100":circuitEndPointSubgroup;

CHARACTERIZED BY

tmCircuitEndPointSubgroupPackage PACKAGE

BEHAVIOUR

tmCircuitEndPointBehaviour BEHAVIOUR

DEFINED AS

"La classe *tmCircuitEndPointSubgroup* est une sous-classe de la classe *circuitEndPointSubgroup* définie dans la Recommandation M.3100. Elle est utilisée pour l'élaboration des commandes et la surveillance des performances à des fins de gestion du trafic.";

ATTRIBUTES

tmSurveillance GET-REPLACE

::

;

REGISTERED AS {q823ObjectClass 26};

11.1.27 tmCircuitEndPointSubgroupCurrentData

tmCircuitEndPointSubgroupCurrentData MANAGED OBJECT CLASS

DERIVED FROM circuitEndPointSubgroupCurrentData;

CHARACTERIZED BY

"ITU-T Rec. X.739:1993":periodSynchronizationPackage,

tmCircuitEndPointSubgroupCurrentDataPackage PACKAGE

BEHAVIOUR

tmCircuitEndPointSubgroupCurrentDataBehaviour BEHAVIOUR

DEFINED AS

"La classe d'objets *tmCircuitEndPointSubgroupCurrentData* est une sous-classe de la classe d'objets *CircuitEndPointSubgroupCurrentData*.

Elle sert à surveiller les données de performances des sous-faisceaux de circuits conformément à la définition de la Recommandation E.502 dans le cadre de la gestion du trafic. Cet objet ou ses sous-classes sont instanciés dans le cadre de la gestion du trafic.

Une instance d'un objet de cette classe ne générera qu'une instance d'un objet de la classe *tmCircuitEndPointSubgroupHistoryData*.

Toutes les instances d'objet de la classe *tmCircuitEndPointSubgroupCurrentData* à l'intérieur du même élément géré auront la même période *granularityPeriod*.

Afin de synchroniser cette période avec la période de temps globale suivante après avoir instancié les sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de *synchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*).";

ATTRIBUTES

"ITU-T Rec. Q.822:1994":historyRetention

PERMITTED VALUES Q823-TM-ASN1Module.PermittedHistoryRetention

GET

::

;

REGISTERED AS {q823ObjectClass 27};

11.1.28 tmExchangeCurrentData

tmExchangeCurrentData MANAGED OBJECT CLASS

DERIVED FROM exchangeCurrentData;

CHARACTERIZED BY

"ITU-T Rec. X.739:1993":periodSynchronizationPackage,

tmExchangeCurrentDataPackage PACKAGE

BEHAVIOUR

tmExchangeCurrentDataBehaviour BEHAVIOUR

DEFINED AS

"La classe d'objets *tmExchangeCurrentData* est une sous-classe de la classe d'objets *ExchangeCurrentData*. Elle sert à surveiller les données de performance de commutateur conformément à la définition de la Recommandation E.502 dans le cadre de la gestion de trafic. Cette classe d'objets ou ses sous-classes sont instanciées à des fins de gestion du trafic.

Une instance d'un objet de cette classe ne générera qu'une instance d'objet de la classe *tmExchangePerformanceHistoryData*.

Afin de synchroniser la période granulaire *granularityPeriod* avec la période de temps globale suivante après avoir instancié les sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de synchronisation de période *periodSynchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*).";

ATTRIBUTES

"ITU-T Rec. Q.822:1994":historyRetention

PERMITTED VALUES Q823-TM-ASN1Module.PermittedHistoryRetention

GET

::

;

REGISTERED AS {q823ObjectClass 28};

11.1.29 tmObservedDestinationCurrentData

tmObservedDestinationCurrentData MANAGED OBJECT CLASS

DERIVED FROM observedDestinationCurrentData;

CHARACTERIZED BY

"ITU-T Rec. X.739:1993":periodSynchronizationPackage,

tmObservedDestinationCurrentDataPackage PACKAGE

BEHAVIOUR

tmObservedDestinationCurrentDataBehaviour BEHAVIOUR

DEFINED AS

"La classe *tmObservedDestinationCurrentData* est une sous-classe de la classe d'objets *observedDestinationCurrentData*.

Elle sert à surveiller les données de performance d'une destination conformément à la définition de la Recommandation E.502 dans le cadre de la gestion de trafic. Cette classe d'objets ou ses sous-classes sont instanciées à des fins de gestion du trafic.

Une instance d'un objet de cette classe ne générera qu'une instance d'objet de la classe *tmObservedDestinationHistoryData*.

Toutes les instances d'objet *tmObservedCurrentData* à l'intérieur du même élément géré auront la même période granulaire.

Afin de synchroniser la période granulaire *granularityPeriod* avec la période de temps globale suivante après l'instance des sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de Synchronisation de période *periodSynchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*).";

ATTRIBUTES

"ITU-T Rec. Q.822:1994":historyRetention

PERMITTED VALUES Q823-TM-ASN1Module.PermittedHistoryRetention

GET

::

;

REGISTERED AS {q823ObjectClass 29};

11.1.30 tmTrafficControlCurrentData

tmTrafficControlCurrentData MANAGED OBJECT CLASS

DERIVED FROM trafficControlCurrentData;

CHARACTERIZED BY

"ITU-T Rec. X.739:1993":periodSynchronizationPackage,

tmTrafficControlCurrentDataPackage PACKAGE

BEHAVIOUR

tmTrafficControlCurrentDataBehaviour BEHAVIOUR

DEFINED AS

"La classe *tmTrafficControlCurrentData* est une sous-classe de la classe *trafficControlCurrentData*.

Elle sert à surveiller l'efficacité des commandes de trafic conformément à la définition de la Recommandation E.502 dans le cadre de la gestion de trafic. Cette classe d'objets ou ses sous-classes sont instanciées à des fins de gestion du trafic.

Une instance d'un objet de cette classe ne générera qu'une instance d'objet de la classe *tmTrafficControlHistoryData*.

Toutes les instances d'objet *tmTrafficControlCurrentData* à l'intérieur du même élément géré auront la même période granulaire.

Afin de synchroniser la période granulaire *granularityPeriod* avec la période de temps globale suivante après l'instance des sous-classes *currentData*, il est recommandé de positionner sur minuit (heure du commutateur) la valeur de synchronisation de période *periodSynchronizationTime* (qui est un attribut du progiciel de synchronisation de période *periodSynchronizationPackage*).";;

ATTRIBUTES

"ITU-T Rec. Q.822:1994":historyRetention

PERMITTED VALUES Q823-TM-ASN1Module.PermittedHistoryRetention

GET

;;

;

REGISTERED AS {q823ObjectClass 30};

11.1.31 trafficControl

trafficControl MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721:1992":top;

CHARACTERIZED BY

"ITU-T Rec. M.3100":objectManagementNotificationsPackage,

trafficControlPackage PACKAGE

BEHAVIOUR

trafficControlBehaviour BEHAVIOUR

DEFINED AS

"La classe *trafficControl* est une "hyperclasse" pour toutes les classes représentant les commandes de trafic comme cela est défini dans la Recommandation E.412. Cette hyperclasse n'est pas instanciée.

Lorsque plusieurs commandes du même type sont instanciées dans la même ressource gérée, un mécanisme (auquel on se rapporte comme clé) est décrit dans chaque sous-classe de la classe *trafficControl* afin de choisir une instance de commande. La commande au critère le plus spécifique a la priorité.";;

ATTRIBUTES

trafficControlId GET,

tmSurveillance GET-REPLACE;

;;

CONDITIONAL PACKAGES

creatorPackage

PRESENT IF "si une instance le prend en charge";

REGISTERED AS {q823ObjectClass 31};

11.1.32 trafficControlCurrentData

trafficControlCurrentData MANAGED OBJECT CLASS

DERIVED FROM "ITU-T Rec. Q.822:1994":currentData;

CHARACTERIZED BY

trafficControlCurrentDataPackage PACKAGE

BEHAVIOUR

trafficControlCurrentDataBehaviour BEHAVIOUR

DEFINED AS

"La classe d'objets *trafficControlCurrentData* est une sous-classe de la classe d'objets *currentData* définie dans la Recommandation Q.822.

Elle sert à surveiller l'efficacité d'une commande de trafic conformément à la définition de la Recommandation E.502.

Cette classe doit être indiquée pour chaque instance de commande. S'il est nécessaire de la préciser avec la classe d'objets de commande, ou d'indiquer l'incidence d'une commande sur une entité gérée, il est possible de synthétiser ces informations au niveau du système d'exploitation à partir de l'instance de commande correspondante.

A des fins de gestion du trafic, on n'instanciera pas cette classe d'objets mais sa sous-classe définie dans la présente Recommandation ou toute autre sous-classe qui en dérive.";;

ATTRIBUTES

callsAffectedByTrafficControl
INITIAL VALUE Q823-TM-ASN1Module.initialCount;
;;

CONDITIONAL PACKAGES

callsOfferedPackage
PRESENT IF "si une instance le prend en charge";
REGISTERED AS {q823ObjectClass 32};

11.1.33 trafficControlHistoryData

trafficControlHistoryData MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Rec. Q.822:1994":historyData;
CHARACTERIZED BY

trafficControlHistoryDataPackage PACKAGE

BEHAVIOUR

trafficControlHistoryDataBehaviour BEHAVIOUR

DEFINED AS

"La classe *trafficControlHistoryData* est une sous-classe de la classe d'objets *historyData* définie dans la Recommandation Q.822.

Les attributs de surveillance des performances sont les mêmes que ceux de la classe *currentData* correspondante.";;

ATTRIBUTES

callsAffectedByTrafficControl GET;
;;

CONDITIONAL PACKAGES

callsOfferedHistoryPackage
PRESENT IF "Ces progiciels sont présents s'ils le sont dans l'instance correspondante de *currentData*";
REGISTERED AS {q823ObjectClass 33};

11.2 Définition de corrélation de nom

La corrélation de nom de la classe *HistoryData* avec la classe *currentData* est définie dans la Recommandation Q.822. Elle est utilisée pour les sous-classes *currentData* et *historyData* définies dans la présente Recommandation.

accAffectedTraffic-managedElement NAME BINDING

SUBORDINATE OBJECT CLASS accAffectedTraffic;

NAMED BY

SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;

WITH ATTRIBUTE accAffectedTrafficId;

BEHAVIOUR

accAffectedTrafficContainmentBehaviour BEHAVIOUR

DEFINED AS "L'opération de suppression échouera si l'une ou plusieurs des instances ACC pointent vers cette instance.";;

CREATE WITH-REFERENCE-OBJECT,

WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE;

REGISTERED AS {q823NameBinding 1};

accTrigger-managedElement NAME BINDING

SUBORDINATE OBJECT CLASS accTrigger AND SUBCLASSES;

NAMED BY

SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;

WITH ATTRIBUTE accTriggerId;
REGISTERED AS {q823NameBinding 2};

adcTrigger-managedElement NAME BINDING
SUBORDINATE OBJECT CLASS adcTrigger AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE adcTriggerId;
CREATE WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE;
REGISTERED AS {q823NameBinding 3};

autoHTRDestination-managedElement NAME BINDING
SUBORDINATE OBJECT CLASS htrDestination AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100": managedElement AND SUBCLASSES;
WITH ATTRIBUTE htrDestinationId;
BEHAVIOUR
autoHTRDestinationBehaviour BEHAVIOUR
DEFINED AS "Cette corrélation de nom est utilisée quand l'agent crée une instance de la classe
htrDestination ou de ses sous-classes, après avoir déterminé, à l'aide de moyens locaux, qu'une
destination est difficile à atteindre.";;
REGISTERED AS {q823NameBinding 10};

circuitEndPointSubgroupHistoryData-tmCircuitEndPointSubgroup NAME BINDING
SUBORDINATE OBJECT CLASS circuitEndPointSubgroupHistoryData AND
SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS tmCircuitEndPointSubgroup AND
SUBCLASSES;
WITH ATTRIBUTE "ITU-T Rec. Q.822":historyDataId;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS {q823NameBinding 4};

congestionLevelIndication-managedElement NAME BINDING
SUBORDINATE OBJECT CLASS congestionLevelIndication AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE congestionLevelIndicationId;
BEHAVIOUR
congestionLevelIndicationToMEBehaviour BEHAVIOUR
DEFINED AS "l'élément de réseau crée cette classe d'objets implicitement.";;
REGISTERED AS {q823NameBinding 5};

dccGroup-managedElement NAME BINDING
SUBORDINATE OBJECT CLASS dccGroup AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE dccGroupId;
BEHAVIOUR
dccGroupContainmentBehaviour BEHAVIOUR
DEFINED AS
"L'opération de suppression échouera si l'une ou plusieurs des instances *destinationCodeControl* pointent
vers cette instance.";;
CREATE WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE;
REGISTERED AS {q823NameBinding 6};

exchangeHistoryData-managedElement NAME BINDING

**SUBORDINATE OBJECT CLASS exchangeHistoryData AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE "ITU-T Rec. Q.822:1994":historyDataId;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS {q823NameBinding 8};**

htrDestination-managedElement NAME BINDING

**SUBORDINATE OBJECT CLASS htrDestination AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE htrDestinationId;
BEHAVIOUR
htrDestinationToMEBehaviour BEHAVIOUR
DEFINED AS "cette corrélation de nom sert quand l'objet est explicitement créé par le gestionnaire.";;
CREATE WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE;
REGISTERED AS {q823NameBinding 9};**

observedDestination-managedElement NAME BINDING

**SUBORDINATE OBJECT CLASS observedDestination AND SUBCLASSES ;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE observedDestinationId;
CREATE WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS {q823NameBinding 11};**

observedDestinationHistoryData-observedDestination NAME BINDING

**SUBORDINATE OBJECT CLASS observedDestinationHistoryData AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS observedDestination AND SUBCLASSES;
WITH ATTRIBUTE "ITU-T Rec.Q.822:1994":historyDataId;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS {q823NameBinding 12};**

scrAffectedTraffic-managedElement NAME BINDING

**SUBORDINATE OBJECT CLASS scrAffectedTraffic AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE scrAffectedTrafficId;
BEHAVIOUR
scrAffectedContainmentBehaviour BEHAVIOUR
DEFINED AS "L'opération de suppression échouera si l'une ou plusieurs des instances SCR
pointent vers cette instance.";;
CREATE WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS {q823NameBinding 13};**

simpleScanner-managedElement NAME BINDING

**SUBORDINATE OBJECT CLASS "ITU-T Rec. X.738:1993":simpleScanner AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":
managedElement AND SUBCLASSES;
WITH ATTRIBUTE "ITU-T Rec. X.739:1993":scannerId;
CREATE WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;**

DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {q823NameBinding 14};

stateIndicator-managedElement NAME BINDING
SUBORDINATE OBJECT CLASS stateIndicator AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE stateIndicatorId;
BEHAVIOUR
stateIndicatorContainmentBehaviour BEHAVIOUR
DEFINED AS "l'élément de réseau crée cette classe d'objets implicitement.";;
REGISTERED AS {q823NameBinding 15};

tmCircuitEndPointSubgroupCurrentData-tmCircuitEndPointSubgroup NAME BINDING
SUBORDINATE OBJECT CLASS tmCircuitEndPointSubgroupCurrentData AND
SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS tmCircuitEndPointSubgroup AND SUBCLASSES;
WITH ATTRIBUTE "ITU-T Rec. X.739:1993":scannerId;
BEHAVIOUR
tmCircuitEndPointSubgroupCurrentDataContainmentBehaviour BEHAVIOUR
DEFINED AS "une instance de la classe *tmCircuitEndPointSubgroupCurrentData* sera
automatiquement instanciée si l'attribut *tmSurveillance* de la classe d'objets
***tmCircuitEndPointSubgroup* supérieure a la valeur Vrai. Une telle instance sera automatiquement**
supprimée si le même attribut *tmSurveillance* a la valeur Faux. L'élément de réseau peut aussi créer
des instances de cette classe implicitement.";;
REGISTERED AS {q823NameBinding 16};

-- NOTE – Les valeurs des attributs de la classe *currentData* sont déterminées par l'élément de réseau
-- à l'aide de moyens locaux.

tmExchangeCurrentData-managedElement NAME BINDING
SUBORDINATE OBJECT CLASS tmExchangeCurrentData AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE "ITU-T Rec. X.739:1993":scannerId;
BEHAVIOUR
tmExchangeCurrentDataContainmentBehaviour BEHAVIOUR
DEFINED AS "seule une instance de la classe *tmExchangeCurrentData* est autorisée dans l'instance
***managedElement*. L'élément de réseau la crée implicitement.";;**
CREATE
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS {q823NameBinding 17};

tmObservedDestinationCurrentData-observedDestination NAME BINDING
SUBORDINATE OBJECT CLASS tmObservedDestinationCurrentData AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS observedDestination AND SUBCLASSES;
WITH ATTRIBUTE "ITU-T Rec. X.739:1993":scannerId;
BEHAVIOUR
tmObservedDestinationCurrentDataContainmentBehaviour BEHAVIOUR
DEFINED AS "une instance de la classe *tmObservedDestinationCurrentData* sera automatiquement
instanciée si l'attribut *tmSurveillance* de la classe d'objets *ObservedDestination* supérieure a la valeur Vrai.
Une telle instance sera automatiquement supprimée si le même attribut *tmSurveillance* a la valeur Faux.
L'élément de réseau peut aussi créer des instances de cette classe implicitement.";;
REGISTERED AS {q823NameBinding 18};

-- NOTE – Les valeurs des attributs de la classe *currentData* sont déterminées par l'élément de réseau
-- à l'aide de moyens locaux.

tmTrafficControlCurrentData-trafficControl NAME BINDING
 SUBORDINATE OBJECT CLASS tmTrafficControlCurrentData AND SUBCLASSES;
 NAMED BY
 SUPERIOR OBJECT CLASS trafficControl AND SUBCLASSES;
 WITH ATTRIBUTE "ITU-T Rec. X.739:1993":scannerId;
 BEHAVIOUR
 tmTrafficControlCurrentDataContainmentBehaviour BEHAVIOUR
 DEFINED AS "une instance de la classe *tmTrafficControlCurrentData* sera automatiquement instanciée si l'attribut *tmSurveillance* de la classe d'objets *trafficControl* supérieure, a la valeur Vrai. Une telle instance sera automatiquement supprimée si le même attribut *tmSurveillance* a la valeur Faux. L'élément de réseau peut aussi créer des instances de cette classe implicitement.";;
 REGISTERED AS {q823NameBinding 19};

-- NOTE – Les valeurs des attributs de la classe *currentData* sont déterminées par l'élément de réseau
 -- à l'aide de moyens locaux.

trafficControl-managedElement NAME BINDING
 SUBORDINATE OBJECT CLASS trafficControl AND SUBCLASSES;
 NAMED BY
 SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElement AND SUBCLASSES;
 WITH ATTRIBUTE trafficControlId;
 CREATE WITH-REFERENCE-OBJECT,
 WITH-AUTOMATIC-INSTANCE-NAMING;
 DELETE DELETES-CONTAINED-OBJECTS;
 REGISTERED AS {q823NameBinding 20};

trafficControlHistoryData-trafficControl NAME BINDING
 SUBORDINATE OBJECT CLASS trafficControlHistoryData AND SUBCLASSES;
 NAMED BY
 SUPERIOR OBJECT CLASS trafficControl AND SUBCLASSES;
 WITH ATTRIBUTE "ITU-T Rec. Q.822:1994":historyDataId;
 DELETE DELETES-CONTAINED-OBJECTS;
 REGISTERED AS {q823NameBinding 21};

11.3 Définition des progiciels

adcAsynchronousTimerPackage PACKAGE
 BEHAVIOUR
 adcAsynchronousTimerBehaviour BEHAVIOUR
 DEFINED AS
 "identifie le temporisateur asynchrone qui spécifie un temps. Ce temporisateur est activé lorsque la tentative d'appel est autorisée; aucune autre tentative n'est autorisée avant son expiration.";;
 ATTRIBUTES
 asynchronousTimer GET;
 REGISTERED AS {q823Package 1};

adcContinuousTimerPackage PACKAGE
 BEHAVIOUR
 adcContinuousTimerBehaviour BEHAVIOUR
 DEFINED AS
 "identifie le temporisateur continu qui comporte le nombre d'appels et une temporisation. Une fois ce nombre de tentatives d'appels traités sur un cycle du temporisateur, aucune autre tentative n'est autorisée avant l'expiration de la temporisation (5 appels en 60 secondes, par exemple).";;
 ATTRIBUTES
 continuousTimer GET;
 REGISTERED AS {q823Package 2};

adcLeakyBucketPackage PACKAGE
 BEHAVIOUR
 adcLeakyBucketTimerBehaviour BEHAVIOUR

DEFINED AS

"identifie le compteur à fuite qui comporte la taille du compteur (valeur maximale autorisée pour le compteur) et le pas de décrémentation par unité de temps. Si la valeur de comptage dépasse la taille maximale définie, la tentative d'appel est annulée. Si cette valeur est inférieure à la taille, la tentative est autorisée et le compteur est incrémenté. Celui-ci est décrément à intervalles de temps définis, ce qui rend l'acceptation de nouveaux appels possible.";;

ATTRIBUTES

leakyBucket GET;

REGISTERED AS {q823Package 3};

adcPercentagePackage PACKAGE**BEHAVIOUR**

adcPercentageBehaviour BEHAVIOUR

DEFINED AS

"identifie l'effet de la commande en pourcentage. Il est prévu de ne prendre en charge qu'un seul choix de pourcentage.";;

ATTRIBUTES

percentage GET;

REGISTERED AS {q823Package 4};

answeredIncomingSeizuresPackage PACKAGE**BEHAVIOUR**

answeredIncomingSeizuresPackageBehaviour BEHAVIOUR

DEFINED AS

"identifie le nombre de prises entrantes lorsqu'un signal de réponse a été transmis vers l'arrière au commutateur précédent.";;

ATTRIBUTES

answeredIncomingSeizures INITIAL VALUE Q823-TM-ASN1Module.initialCount;

REGISTERED AS {q823Package 5};

answeredIncomingSeizureHistoryPackage PACKAGE**BEHAVIOUR**

answeredIncomingSeizureHistoryPackageBehaviour BEHAVIOUR

DEFINED AS

"identifie le nombre de prises entrantes lorsqu'un signal de réponse a été transmis vers l'arrière au commutateur précédent.";;

ATTRIBUTES

answeredIncomingSeizures GET;

REGISTERED AS {q823Package 6};

assocOwnerDccGroupPackage PACKAGE**BEHAVIOUR**

assocOwnerDccGroupBehaviour BEHAVIOUR

DEFINED AS

"pointe vers l'instance dccGroup.";;

ATTRIBUTES

assocOwnerDccGroup GET-REPLACE;

REGISTERED AS {q823Package 7};

asynchronousTimerPackage PACKAGE**BEHAVIOUR**

asynchronousTimerPackageBehaviour BEHAVIOUR

DEFINED AS

"identifie le temporisateur asynchrone qui spécifie un temps. Ce temporisateur est activé lorsque la tentative d'appel est autorisée; aucune autre tentative n'est autorisée avant l'expiration de la temporisation.";;

ATTRIBUTES

asynchronousTimer GET-REPLACE;

REGISTERED AS {q823Package 8};

callsAffectedByDccPackage PACKAGE
BEHAVIOUR
callsAffectedByDccPackageBehaviour BEHAVIOUR
DEFINED AS
 "identifie le nombre d'appels de la destination observée qui ont été affectés par la fonction de commande de destination ventilée par type de commande.";;
ATTRIBUTES
callsAffectedByDcc INITIAL VALUE Q823-TM-ASN1Module.initialCount;
REGISTERED AS {q823Package 9};
callsAffectedByDccPackage PACKAGE
BEHAVIOUR
callsAffectedByDccPackageBehaviour BEHAVIOUR
DEFINED AS
 "identifie le nombre d'appels de la destination observée et affectés par les commandes de destination.";;
ATTRIBUTES
callsAffectedByDcc GET;
REGISTERED AS {q823Package 10};

callsOfferedPackage PACKAGE
BEHAVIOUR
callsOfferedBehaviour BEHAVIOUR
DEFINED AS
 "identifie le nombre d'appels offerts à l'instance commande de trafic.";;
ATTRIBUTES
callsOfferedToTrafficControl INITIAL VALUE Q823-TM-ASN1Module.initialCount;
REGISTERED AS {q823Package 11};

callsOfferedHistoryPackage PACKAGE
BEHAVIOUR
callsOfferedHistoryBehaviour BEHAVIOUR
DEFINED AS
 "identifie le nombre d'appels offerts à l'instance commande de trafic.";;
ATTRIBUTES
callsOfferedToTrafficControl GET;
REGISTERED AS {q823Package 12};

continuousTimerPackage PACKAGE
BEHAVIOUR
continuousTimerPackageBehaviour BEHAVIOUR
DEFINED AS
 "identifie le temporisateur continu qui comporte le nombre d'appels et une temporisation. Une fois ce nombre de tentatives d'appels traités sur un cycle du temporisateur, aucune autre tentative n'est autorisée avant l'expiration de la temporisation (5 appels en 60 secondes, par exemple).";;
ATTRIBUTES
continuousTimer GET-REPLACE;
REGISTERED AS {q823Package 13};

creatorPackage PACKAGE
BEHAVIOUR
creatorBehaviour BEHAVIOUR
DEFINED AS
 "identifie le créateur de l'instance de l'objet. La valeur de cet attribut sera positionnée par l'agent au moment de la création. Les moyens de détermination de l'identité du créateur sont locaux.";;
ATTRIBUTES
creatorIdentity GET;
REGISTERED AS {q823Package 14};

destinationCodePackage PACKAGE
BEHAVIOUR
destinationCodePackageBehaviour BEHAVIOUR

DEFINED AS
 "identifie la destination.";;

ATTRIBUTES
 destinationCode GET;

REGISTERED AS {q823Package 15};

destinationTypePackage PACKAGE
BEHAVIOUR
 destinationTypePackageBehaviour BEHAVIOUR

DEFINED AS
 "identifie le type de destination. Il s'agit de la nature de l'adresse exprimée par une chaîne de 7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste de valeurs de type énuméré.";;

ATTRIBUTES
 destinationType GET;

REGISTERED AS {q823Package 16};

leakyBucketPackage PACKAGE
BEHAVIOUR
 leakyBucketTimerBehaviour BEHAVIOUR

DEFINED AS
 "identifie le compteur à fuite qui comporte la taille du compteur (valeur maximale autorisée pour le compteur) et le pas de décrémentation par unité de temps. Si la valeur de comptage dépasse la taille maximale définie, la tentative d'appel est annulée. Si cette valeur est inférieure ou égale à la taille maximale, la tentative est autorisée et le compteur est incrémenté. Celui-ci est décrémenté à intervalles de temps définis ce qui rend l'acceptation de nouveaux appels possible.";;

ATTRIBUTES
 leakyBucket GET-REPLACE;

REGISTERED AS {q823Package 17};

level2ResponseCategoriesPackage PACKAGE
BEHAVIOUR
 level2ResponseCategoriesPackageBehaviour BEHAVIOUR

DEFINED AS
 "identifie dans une séquence l'effet d'une combinaison des aspects liés à l'acheminement, l'origine ou la destination ou d'autres critères correspondant au 2^{ème} niveau d'encombrement.";;

ATTRIBUTES
 level2ResponseCategories GET-REPLACE;

REGISTERED AS {q823Package 18};

percentagePackage PACKAGE
BEHAVIOUR
 percentagePackageBehaviour BEHAVIOUR

DEFINED AS
 "identifie l'effet de la commande en pourcentage. Il est prévu de ne prendre en charge qu'un seul de ces choix.";;

ATTRIBUTES
 percentage GET-REPLACE;

REGISTERED AS {q823Package 19};

stateIndicatorActionPackage PACKAGE
BEHAVIOUR
 stateIndicatorActionPackageBehaviour BEHAVIOUR

DEFINED AS "afin de déclencher l'émission de la réponse ACTION REPLY contenant la phototrame montrant les conditions de l'élément de réseau pendant la période *granularityPeriod*.";;

ACTIONS
 stateIndicatorAction;

REGISTERED AS {q823Package 20};

stateIndicatorNotificationPackage PACKAGE
 BEHAVIOUR
 stateIndicatorNotificationPackageBehaviour BEHAVIOUR
 DEFINED AS "cet avis est automatiquement émis à la fin de la période *granularityPeriod*, et fournit une phototrame montrant les conditions de l'élément de réseau durant la période *granularityPeriod*.";;
 NOTIFICATIONS
 stateIndicatorNotification;
 REGISTERED AS {q823Package 21};

tmCircuitEndPointSubgroupListPackage PACKAGE
 BEHAVIOUR
 tmCircuitEndPointSubgroupListPackageBehaviour BEHAVIOUR
 DEFINED AS
 "identifie la liste des sous-faisceaux de circuits";;
 ATTRIBUTES
 tmCircuitEndPointSubgroupList
 GET-REPLACE
 ADD-REMOVE;
 REGISTERED AS {q823Package 22};

triggerThresholdPackage PACKAGE
 BEHAVIOUR
 triggerThresholdPackageBehaviour BEHAVIOUR
 DEFINED AS
 "identifie la valeur du seuil de déclenchement et instanciera la commande ADC si cette valeur est dépassée.";;
 ATTRIBUTES
 triggerThreshold GET;
 REGISTERED AS {q823Package 23};

11.4 Définition du comportement commun

instancePointerAndNameBehaviour BEHAVIOUR
 DEFINED AS "si on choisit la syntaxe de type chaîne, il est permis de faire correspondre les deux sous-chaînes. Si on choisit la syntaxe de type nombre, il est alors permis d'effectuer une correspondance de type relation d'ordre.";

11.5 Definition of attributes

accAffectedTrafficId ATTRIBUTE
 DERIVED FROM trafficManagementObjectRdn;
 REGISTERED AS {q823Attribute 1};

accResponseCategories ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.ResponseCategories;
 BEHAVIOUR
 accResponseValueBehaviour BEHAVIOUR
 DEFINED AS
 "identifie dans une séquence, l'effet en pourcentage qui correspond à une combinaison des aspects liés à l'acheminement, la destination, l'origine ou d'autres critères.";;
 REGISTERED AS {q823Attribute 2};

activationThresholds ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.ActivationThresholds;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
 activationThresholdBehaviour BEHAVIOUR
 DEFINED AS "identifie les seuils d'activation de la commande.";;
 REGISTERED AS {q823Attribute 3};

accTriggerId **ATTRIBUTE**
 DERIVED FROM trafficManagementObjectRdn;
REGISTERED AS {q823Attribute 4};

adcTriggerId **ATTRIBUTE**
 DERIVED FROM trafficManagementObjectRdn;
REGISTERED AS {q823Attribute 5};

adcTriggerType **ATTRIBUTE**
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.AdcTriggerType;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
 adcTriggerTypeBehaviour BEHAVIOUR
 DEFINED AS
 "identifie si l'instance d'objet commande adc créée est 'centralized', 'decentralized' ou 'both'. '";
REGISTERED AS {q823Attribute 6};

adcType **ATTRIBUTE**
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.AdcType;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
 adcTypeBehaviour BEHAVIOUR
 DEFINED AS
 "identifie si l'instance d'objet commande adc créée est 'centralized' ou 'decentralized'. '";
REGISTERED AS {q823Attribute 7};

answer **ATTRIBUTE**
 DERIVED FROM "CCITT Rec. X.721:1992":counter;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
 answerBehaviour BEHAVIOUR
 DEFINED AS
 "information de réponse d'appel envoyée dans l'indication émise vers l'arrière. Ce terme est défini dans
 l'Annexe A/E.410.'";
REGISTERED AS {q823Attribute 8};

answeredIncomingSeizures **ATTRIBUTE**
 DERIVED FROM answer;
 BEHAVIOUR
 answeredIncomingSeizuresBehaviour BEHAVIOUR
 DEFINED AS
 "identifie le nombre de prises entrantes lorsqu'un signal de réponse est émis vers l'arrière en direction du
 commutateur précédent.'";
REGISTERED AS {q823Attribute 9};

answeredOutgoingSeizures **ATTRIBUTE**
 DERIVED FROM answer;
 BEHAVIOUR
 answeredOutgoingSeizuresBehaviour BEHAVIOUR
 DEFINED AS
 "identifie le nombre de prises sortantes lors de la réception d'un signal de réponse.'";
REGISTERED AS {q823Attribute 10};

assocAccAffectedTraffic **ATTRIBUTE**
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.AssocAccAffectedTraffic;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
 assocAccAffectedTrafficBehaviour BEHAVIOUR

DEFINED AS
 "pointe vers l'instance *accAffectedTraffic* pour les niveaux d'encombrement 1 et 2.";;
REGISTERED AS {q823Attribute 66};

assocOwnerDccGroup **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.AssocOwnerDccGroup;
MATCHES FOR EQUALITY;
BEHAVIOUR
assocOwnerDccGroupPointerBehaviour **BEHAVIOUR**
DEFINED AS
 "pointe vers l'instance d'objet *dccGroup*.";;
REGISTERED AS {q823Attribute 11};

assocScrAffectedTraffic **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.AssocScrAffectedTraffic;
MATCHES FOR EQUALITY;
BEHAVIOUR
assocScrAffectedTrafficBehaviour **BEHAVIOUR**
DEFINED AS
 "pointe vers l'instance *scrAffectedTraffic* pour les niveaux de réservation 1 et 2.";;
REGISTERED AS {q823Attribute 53};

asynchronousTimer **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.Timer;
MATCHES FOR EQUALITY;
BEHAVIOUR
asynchronousTimerBehaviour **BEHAVIOUR**
DEFINED AS
 "identifie le temporisateur asynchrone qui spécifie un temps. Le temporisateur est activé lorsque la tentative d'appel est autorisée; aucune autre tentative n'est autorisée avant l'expiration de la temporisation. Il est prévu de ne prendre en charge qu'un choix de temporisation.";;
REGISTERED AS {q823Attribute 12};

autoActivated **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.TrueFalse;
MATCHES FOR EQUALITY;
BEHAVIOUR
autoActivatedBehaviour **BEHAVIOUR**
DEFINED AS
 "indique si un mécanisme de déclenchement est actif pour cette commande. Si l'attribut *autoActivated* a la valeur 'TRUE' et si l'attribut *administrativeState* a la valeur 'unlocked', la commande automatique est activée. Cette commande est désactivée pour toutes les autres combinaisons de valeurs des attributs *autoActivated* et *administrativeState*. La valeur de cet attribut initialisée et tenue à jour par l'élément de réseau.";;
REGISTERED AS {q823Attribute 13};

bids **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
MATCHES FOR EQUALITY;
BEHAVIOUR
bidsBehaviour **BEHAVIOUR**
DEFINED AS
 "tentative d'obtention d'un circuit d'un sous-faisceau ou d'une destination. La tentative de prise peut être réussie ou manquée. Ce terme est défini à l'Annexe A/E.410.";;
REGISTERED AS {q823Attribute 14};

callsAffectedByDcc **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
MATCHES FOR EQUALITY;

BEHAVIOUR
callsAffectedByDccBehaviour **BEHAVIOUR**
DEFINED AS
 "identifie le nombre d'appels de la destination observée qui sont affectés par la commande d'indicatif de destination.";;
REGISTERED AS {q823Attribute 15};

callsAffectedByTrafficControl **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
callsAffectedByTrafficControlBehaviour **BEHAVIOUR**
DEFINED AS
 "identifie le nombre d'appels offerts à l'instance commande de trafic.";;
REGISTERED AS {q823Attribute 16};

callsBlockedByLoadShedding **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
callsBlockedByLoadSheddingBehaviour **BEHAVIOUR**
DEFINED AS
 "identifie le nombre d'appels qui ne peuvent être traités à cause d'un mécanisme de protection interne au commutateur contre la surcharge.";;
REGISTERED AS {q823Attribute 17};

callsOfferedToTrafficControl **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
callsOfferedToTrafficControlBehaviour **BEHAVIOUR**
DEFINED AS
 "identifie le nombre d'appels offerts à l'instance commande de trafic.";;
REGISTERED AS {q823Attribute 18};

c11ResponseCategories **ATTRIBUTE**
WITH **ATTRIBUTE** **SYNTAX** **Q823-TM-**
ASN1Module.ResponseCategories;
MATCHES FOR EQUALITY;
BEHAVIOUR
c11ResponseCategoriesBehaviour **BEHAVIOUR**
DEFINED AS
 "catégorie de réponses du 1^{er} niveau d'encombrement.";;
REGISTERED AS {q823Attribute 19};

c12ResponseCategories **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module. ResponseCategories;
MATCHES FOR EQUALITY;
BEHAVIOUR
c12ResponseCategoriesBehaviour **BEHAVIOUR**
DEFINED AS
 catégorie de réponses du 2^{ème} niveau d'encombrement.";;
REGISTERED AS {q823Attribute 20};

congestionLevel **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.CongestionLevel;
MATCHES FOR EQUALITY;
BEHAVIOUR
congestionLevelBehaviour **BEHAVIOUR**
DEFINED AS
 "indique l'état d'encombrement d'un commutateur. Il est exprimé par les niveaux d'encombrement de machine (MCL) suivants:
 – MCL0: pas d'encombrement du commutateur
 – MCL1: encombrement modéré du commutateur; celui-ci continue à travailler. Certains appels peuvent être rejetés."

destinationCode **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.DestinationCode;
MATCHES FOR EQUALITY, SUBSTRINGS, ORDERING;
BEHAVIOUR
destinationCodeBehaviour **BEHAVIOUR**
DEFINED AS
"identifie l'indicatif de pays ou de zone ou le code d'identification d'un commutateur ou le
numéro d'un autre lieu auquel s'applique l'instance d'objet.";;
REGISTERED AS {q823Attribute 27};

destinationType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.DestinationType;
MATCHES FOR EQUALITY;
BEHAVIOUR
destinationTypeBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le type de destination. Il s'agit de la nature de l'adresse exprimée par une chaîne de
7 bits conformément à la Recommandation Q.763 ou du type de destination exprimé par une liste
de valeurs de type énuméré.";;
REGISTERED AS {q823Attribute 28};

dispositionOfCalls **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.DispositionOfCall;
MATCHES FOR EQUALITY;
BEHAVIOUR
dispositionOfCallBehaviour **BEHAVIOUR**
DEFINED AS
"identifie si les appels sont déviés vers le prochain sous-faisceau disponible (si la valeur est
Néant) ou annulés (si la valeur est traitement).";;
REGISTERED AS {q823Attribute 29};

htrDestinationId **ATTRIBUTE**
DERIVED FROM trafficManagementObjectRdn;
REGISTERED AS {q823Attribute 30};

incomingSeizures **ATTRIBUTE**
DERIVED FROM seizure;
BEHAVIOUR
incomingSeizuresBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le nombre de prises entrantes sur le sous-faisceau.";;
REGISTERED AS {q823Attribute 31};

incomingTraffic **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
incomingTrafficBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le nombre d'appels entrants dans le commutateur pour lesquels il a été accusé
réception d'au moins un des numéros.";;
REGISTERED AS {q823Attribute 32};

incomingTrafficUsage **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.Integer;
BEHAVIOUR
incomingTrafficUsageBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le trafic entrant écoulé en erlang. Il est d'ordinaire déterminé avec une méthode
d'échantillonnage.";;
REGISTERED AS {q823Attribute 33};

internalTraffic ATTRIBUTE
 DERIVED FROM "CCITT Rec. X.721:1992":counter;
 BEHAVIOUR
internalTrafficBehaviour BEHAVIOUR
 DEFINED AS
 "identifie le nombre d'appels internes (prises) effectués par le commutateur.";;
 REGISTERED AS {q823Attribute 34};

leakyBucket ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.LeakyBucket;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
leakyBucketBehaviour BEHAVIOUR
 DEFINED AS
 "identifie le compteur à fuite qui comporte la taille du compteur (valeur maximale autorisée pour le compteur) et le pas de décrémentation par unité de temps. Si la valeur de comptage dépasse la taille maximale définie, la tentative d'appel est annulée. Si cette valeur est inférieure ou égale à la taille maximale, la tentative est autorisée et le compteur est incrémenté. Celui-ci est décrémenté à intervalles de temps définis ce qui rend l'acceptation de nouveaux appels possible. L'élément de réseau fournira la taille *bucketSize* si elle n'est pas fournie. Il est prévu de ne prendre en charge qu'un choix de temporisation.";;
 REGISTERED AS {q823Attribute 35};

level1ResponseCategories ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.ResponseCategories;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
level1ResponseCategoriesBehaviour BEHAVIOUR
 DEFINED AS
 "catégories de réponses du 1^{er} niveau de réservation.";;
 REGISTERED AS {q823Attribute 36};

level2ResponseCategories ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.ResponseCategories;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
level2ResponseCategoriesBehaviour BEHAVIOUR
 DEFINED AS
 "catégories de réponses du 2^{ème} niveau de réservation.";;
 REGISTERED AS {q823Attribute 37};

newTmCircuitEndPointSubgroups ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.NewTmCircuitEndPointSubgroups;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
newTmCircuitEndPointSubgroupsBehaviour BEHAVIOUR
 DEFINED AS
 "identifie dans une séquence les sous-faisceaux qui sont au repos afin d'établir les appels.";;
 REGISTERED AS {q823Attribute 38};

noCircuitsAvailable ATTRIBUTE
 DERIVED FROM "CCITT Rec. X.721:1992":counter;
 BEHAVIOUR
noCircuitsAvailableBehaviour BEHAVIOUR
 DEFINED AS
 "identifie le nombre de tentatives aboutissant à des communications inefficaces parce qu'il n'existe aucun circuit libre et disponible vers la destination observée; c'est-à-dire que l'on a un débordement sur le dernier sous-faisceau de circuits correspondant à cette destination.";;
 REGISTERED AS {q823Attribute 39};

numberOfAvailCircuits ATTRIBUTE
DERIVED FROM "CCITT Rec. X.721:1992":gauge;
BEHAVIOUR
numberOfAvailCircuitsBehaviour BEHAVIOUR
DEFINED AS
"identifie le nombre de circuits capables d'écouler du trafic y compris ceux qui sont en train d'en écouler.
Le fait qu'il s'agisse d'une valeur instantanée ou moyenne est laissé à la discrétion du fabricant à cause
des changements normalement très fréquents sur les circuits. Les deux méthodes sont équivalentes.";;
REGISTERED AS {q823Attribute 40};

observedDestinationId ATTRIBUTE
DERIVED FROM trafficManagementObjectRdn;
REGISTERED AS {q823Attribute 41};

originationAspect ATTRIBUTE
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.OriginationAspect;
MATCHES FOR EQUALITY;
BEHAVIOUR
instancePointerAndNameBehaviour,
originationAspectBehaviour BEHAVIOUR
DEFINED AS
"identifie la catégorie de l'origine et de l'appelant (conformément à la Recommandation Q.763)
pour lesquelles cette commande est valide. Si la valeur de cet attribut est une séquence vide, la
commande de trafic est valide pour tous les aspects liés à l'origine.";;
REGISTERED AS {q823Attribute 42};

originatingTraffic ATTRIBUTE
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
originatingTrafficBehaviour BEHAVIOUR
DEFINED AS
"identifie le trafic de départ du commutateur c'est-à-dire le nombre d'appels (prises) qui se présentent sur
la chaîne de départ du commutateur et pour lesquels il a été accusé réception d'au moins un numéro.";;
REGISTERED AS {q823Attribute 43};

outgoingBids ATTRIBUTE
DERIVED FROM bids;
BEHAVIOUR
outgoingBidsBehaviour BEHAVIOUR
DEFINED AS
"identifie le nombre de tentatives réussies ou manquées de prise d'un circuit d'un sous-faisceau.";;
REGISTERED AS {q823Attribute 44};

outgoingSeizures ATTRIBUTE
DERIVED FROM seizure;
BEHAVIOUR
outgoingSeizuresBehaviour BEHAVIOUR
DEFINED AS
"identifie le nombre de tentatives de prises sortantes d'un des circuits d'un sous-faisceau qui ont réussi.";;
REGISTERED AS {q823Attribute 45};

outgoingTraffic ATTRIBUTE
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
outgoingTrafficBehaviour BEHAVIOUR
DEFINED AS
"identifie le nombre d'appels sortants (prises) du commutateur qui ont abouti à une prise de circuit";;
REGISTERED AS {q823Attribute 46};

outgoingTrafficUsage **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.Integer;
BEHAVIOUR
outgoingTrafficUsageBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le trafic sortant écoulé exprimé en erlang. Il est d'ordinaire déterminé avec une méthode d'échantillonnage.";;
REGISTERED AS {q823Attribute 47};

overflow **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
overflowBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le nombre de tentatives de prises sortantes qui débordent de ce sous-faisceau de circuits. Ce nombre ne tient pas compte des appels affectés par les commandes annulation du trafic de débordement réacheminé (*Cancel Rerouted Overflow*), débordement temporaire depuis (*TAR from*), annulation du débordement depuis (*CancelFrom*)";;
REGISTERED AS {q823Attribute 48};

percentage **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.Percentage;
MATCHES FOR EQUALITY;
BEHAVIOUR
percentageBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le pourcentage d'appels annulés suite à l'activation d'une commande. Il est prévu de ne prendre en charge qu'un seul choix de pourcentage pour l'effet de la commande.";;
REGISTERED AS {q823Attribute 49};

returnAction **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.ReturnAction;
MATCHES FOR EQUALITY;
BEHAVIOUR
returnActionBehaviour **BEHAVIOUR**
DEFINED AS
"identifie le traitement du trafic débordant des sous-faisceaux *new CircuitEndPoint-Subgroups*.";;
REGISTERED AS {q823Attribute 50};

routingAspect **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.RoutingAspect;
MATCHES FOR EQUALITY;
BEHAVIOUR
routingAspectBehaviour **BEHAVIOUR**
DEFINED AS
"identifie l'aspect lié à l'acheminement (trafic de premier choix ou de débordement) pour lequel cette commande est valide. Si la valeur de l'attribut est Néant, la commande de trafic s'applique à tous les aspects liés à l'acheminement.";;
REGISTERED AS {q823Attribute 51};

scrAffectedTrafficId **ATTRIBUTE**
DERIVED FROM trafficManagementObjectRdn;
REGISTERED AS {q823Attribute 52};

seizure **ATTRIBUTE**
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
seizureBehaviour **BEHAVIOUR**
DEFINED AS
"tentative de prise réussie d'un des circuits d'un sous-faisceau. Ce terme est défini à l'Annexe A/E.410.";;
REGISTERED AS {q823Attribute 54};

```

stateIndicatorId  ATTRIBUTE
                  DERIVED FROM trafficManagementObjectRdn;
REGISTERED AS {q823Attribute 55};

terminatingTraffic  ATTRIBUTE
                   DERIVED FROM "CCITT Rec. X.721:1992":counter;
                   BEHAVIOUR
                   terminatingTrafficBehaviour  BEHAVIOUR
                   DEFINED AS
                   "identifie le nombre d'appels (prises) du commutateur qui arrivent aux joncteurs d'abonnés.";;
REGISTERED AS {q823Attribute 56};

timeExpiration     ATTRIBUTE
                   WITH ATTRIBUTE SYNTAX
                   Q823-TM-ASN1Module.Timer;
                   MATCHES FOR EQUALITY;
                   BEHAVIOUR
                   timeExpiringBehaviour       BEHAVIOUR
                   DEFINED AS
                   "identifie le temps après lequel la commande sera automatiquement supprimée.";;
REGISTERED AS {q823Attribute 57};

tmCircuitEndPointSubgroupList  ATTRIBUTE
                                WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.TmCircuitEndPointSubgroupList;
                                BEHAVIOUR
                                tmCircuitEndPointSubgroupListBehaviour  BEHAVIOUR
                                DEFINED AS
                                "identifie les sous-faisceaux de circuits pour lesquels cette instance d'objet s'applique. Si la valeur de
                                cet attribut est vide, l'instance s'applique à tous les sous-faisceaux.";;
REGISTERED AS {q823Attribute 59};

tmSurveillance  ATTRIBUTE
                WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.TrueFalse;
                MATCHES FOR EQUALITY;
                BEHAVIOUR
                tmSurveillanceBehaviour  BEHAVIOUR
                DEFINED AS
                "indique si on est en train de surveiller la gestion de trafic de l'instance d'objet. Positionner la valeur de
                cet attribut sur Vrai entraînera la création d'une instance de la sous-classe CurrentData nommée à partir
                de cette instance et dont la valeur d'attribut scannerId (attribut de dénomination) est égale à '1'. Si la
                création de l'objet de la classe CurrentData échoue, le positionnement de l'attribut échouera et sa valeur
                restera inchangée. Si l'attribut a déjà la valeur Vrai avant l'opération de positionnement, cette opération
                réussira mais n'aura aucun effet sur l'objet en question.
                Positionner la valeur de cet attribut sur Faux entraînera la suppression d'une instance de la sous-classe
                CurrentData nommée à partir de cette instance d'objet et dont la valeur d'attribut scannerId (attribut de
                dénomination) est égale à '1'. Si la suppression de l'objet de la classe CurrentData échoue, le
                positionnement de l'attribut échouera et sa valeur restera inchangée. Si la valeur de l'attribut était déjà
                positionnée sur Faux avant l'opération de positionnement, cette opération réussira mais n'aura aucun
                effet sur l'objet en question.";;
REGISTERED AS {q823Attribute 60};

trafficControlId  ATTRIBUTE
                  DERIVED FROM trafficManagementObjectRdn;
REGISTERED AS {q823Attribute 61};

trafficManagementObjectRdn  ATTRIBUTE
                             WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.NameType;
                             MATCHES FOR EQUALITY;
                             BEHAVIOUR
                             trafficManagementObjectRdnBehaviour  BEHAVIOUR

```

DEFINED AS
"identificateur d'objet.";;
REGISTERED AS {q823Attribute 62};

transitTraffic ATTRIBUTE
DERIVED FROM "CCITT Rec. X.721:1992":counter;
BEHAVIOUR
transitTrafficBehaviour BEHAVIOUR
DEFINED AS
"identifie le nombre d'appels en transit (prises) du commutateur.";;
REGISTERED AS {q823Attribute 63};

treatment ATTRIBUTE
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.Treatment;
MATCHES FOR EQUALITY;
BEHAVIOUR
instancePointerAndNameBehaviour,
treatmentBehaviour BEHAVIOUR
DEFINED AS
"identifie le traitement du flux de trafic affecté par une annulation (annonce, par exemple).";;
REGISTERED AS {q823Attribute 64};

triggerThreshold ATTRIBUTE
WITH ATTRIBUTE SYNTAX Q823-TM-ASN1Module.IntegerValue;
MATCHES FOR EQUALITY;
BEHAVIOUR
triggerThresholdBehaviour BEHAVIOUR
DEFINED AS
"identifie le seuil de déclenchement qui instanciera automatiquement une commande";;
REGISTERED AS {q823Attribute 65};

11.6 Définition des Actions

stateIndicatorAction ACTION
BEHAVIOUR
stateIndicatorActionBehaviour BEHAVIOUR
DEFINED AS "pour déclencher la réponse ACTION REPLY contenant une phototrame montrant l'état de l'élément de réseau durant la période *granularityPeriod*. La position des bits et leur définition sont les suivantes:
bit-0: *exchangeCongestionLevel1* – 1^{er} niveau d'encombrement du commutateur (CL1): ce bit est égal à 1 si durant la période *granularityPeriod*, l'élément de réseau se trouve au 1^{er} niveau d'encombrement.
bit-1: *exchangeCongestionLevel2* – 2^{ème} niveau d'encombrement du commutateur (CL2): ce bit est égal à 1 si durant la période *granularityPeriod*, l'élément de réseau se trouve au 2^{ème} niveau d'encombrement.
bit-2: *congestionLevel1Received* – indicateur CL1 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL1 d'un des commutateurs adjacents.
bit-3: *congestionLevel2Received* – indicateur CL2 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL2 d'un des commutateurs adjacents.
bit-4: *scrTriggered* – déclenchement SCR: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur 'unlocked' et si l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande SCR.
bit-5: *accTriggered* – déclenchement ACC: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur 'unlocked' et l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande ACC.
bit-6: *protectiveControlActive* – commande protectrice activée: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes protectrices (c'est-à-dire: *cancelTo*, *cancelFrom*, *skip*, *ACC*, *SCR*, *cancelRerouted*) sont présentes dans l'élément de réseau.
bit-7: *expansiveControlActive* – commande expansive activée: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes expansives (c'est-à-dire: *tarTo*, *tarFrom*) sont présentes dans l'élément de réseau.

bit-8: *destinationControlActive* – activation de commande de destination: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes de destination (c'est-à-dire: fonctions *destinationCodeControl*, ADC) sont présentes dans l'élément de réseau.

bit-9: *htrDestinationActive* – activation de commande de destination: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes *htrDestination* sont présentes dans l'élément de réseau.

bit-10: *circuitEndPointSubgroupAddedOrDeleted* – addition ou suppression d'un sous-faisceau d'extrémité de circuits: ce bit est égal à 1 durant la période *granularityPeriod* si un sous-faisceau est ajouté ou supprimé dans l'élément de réseau.

bit-11: *accTransmissionInhibited* – transmission acc inhibée: ce bit est égal à 1 si durant la période *granularityPeriod*, l'attribut *administrativeState* du mécanisme *accTrigger* est à l'état verrouillé;

bit-12: *adcTriggered* – commande adc déclenchée: ce bit est égal à 1 si durant la période *granularityPeriod*, l'attribut *administrativeState* est à l'état déverrouillé et l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande ADC.";

MODE CONFIRMED;

WITH REPLY SYNTAX Q823-TM-ASN1Module.StateIndicatorBitMap;

REGISTERED AS {q823Action 1};

11.7 Définition des avis

stateIndicatorNotification NOTIFICATION

BEHAVIOUR

stateIndicatorNotificationBehaviour BEHAVIOUR

DEFINED AS "Cette notification est émise automatiquement à la fin de la période *granularityPeriod* et fournit une phototrame montrant l'état de l'élément de réseau durant cette période. La position des bits et leur définition sont les suivantes:

bit-0: *exchangeCongestionLevel1* – 1^{er} niveau d'encombrement du commutateur (CL1): ce bit est égal à 1 si durant la période *granularityPeriod*, l'élément de réseau se trouve au 1^{er} niveau d'encombrement.

bit-1: *exchangeCongestionLevel2* – 2^{ème} niveau d'encombrement du commutateur (CL2): ce bit est égal à 1 si durant la période *granularityPeriod*, l'élément de réseau se trouve au 2^{ème} niveau d'encombrement.

bit-2: *congestionLevel1Received* – indicateur CL1 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL1 d'un des commutateurs adjacents.

bit-3: *congestionLevel2Received* – indicateur CL2 reçu: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'élément de réseau reçoit un indicateur CL2 d'un des commutateurs adjacents.

bit-4: *scrTriggered* – déclenchement SCR: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur 'unlocked' et l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande SCR.

bit-5: *AccTriggered* – déclenchement ACC: ce bit est égal à 1 si durant la période granulaire *granularityPeriod*, l'attribut *administrativeState* a la valeur 'unlocked' et l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande ACC.

bit-6: *protectiveControlActive* – commande protectrice activée: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes protectrices (c'est-à-dire: *cancelTo*, *cancelFrom*, *skip*, ACC, SCR, *cancelRerouted*) sont présentes dans l'élément de réseau.

bit-7: *expansiveControlActive* – commande expansive activée: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes expansives (c'est-à-dire: *tarTo*, *tarFrom*) sont présentes dans l'élément de réseau.

bit-8: *destinationControlActive* – activation de commande de destination: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes de destination (c'est-à-dire: fonctions *destinationCodeControl*, ADC) sont présentes dans l'élément de réseau.

bit-9: *htrDestinationActive* – activation de commande de destination: ce bit est égal à 1 si durant la période *granularityPeriod* une ou plusieurs instances de commandes *htrDestination* sont présentes dans l'élément de réseau.

bit-10: *circuitEndPointSubgroupAddedOrDeleted* – addition ou suppression d'un sous-faisceau d'extrémité de circuits: ce bit est égal à 1 durant la période *granularityPeriod* si un sous-faisceau est ajouté ou supprimé dans l'élément de réseau.

bit-11: *accTransmissionInhibited* – transmission acc inhibée: ce bit est égal à 1 si durant la période *granularityPeriod*, l'attribut *administrativeState* du mécanisme *accTrigger* est à l'état verrouillé;

bit-12: *adcTriggered* – commande adc déclenchée: ce bit est égal à 1 si durant la période *granularityPeriod*, l'attribut *administrativeState* est à l'état déverrouillé et l'attribut *autoActivated* a la valeur Vrai pour au moins une instance de commande ADC.";;

WITH INFORMATION SYNTAX Q823-TM-ASN1Module.StateIndicatorBitMap;

REGISTERED AS {q823Notification 1};

11.8 Module des types définis ASN.1

-- règles d'extensibilité

-- conformément à l'ISO 8824-DAM portant sur les règles d'extensibilité, les productions de types

-- extensibles doivent être indiquées en introduisant trois (3) points de suspension (...) dans leur description

-- type.

Q823 – TM-ASN1Module {ccitt(0) recommendation(0) q(17) q823(823) asn1Module(2) q823ASN1 Module(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- *EXPORTER tout*

IMPORTS

**Attribute,
ObjectInstance**

FROM

CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}

AdministrativeState,

Count,

ObservedValue

FROM

Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}

HistoryRetention

FROM

Q822-PM-ASN1Module {ccitt(0) recommendation(0) q(17) q822(822) asn1Module(2) q822ASN1Module(0)}

NameType,

FROM

ASN1DefinedTypesModule {ccitt recommendation m(13) gnm(3100) informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)};

q823-InformationModel OBJECT IDENTIFIER ::= {ccitt(0) recommendation(0) q(17) q823(823) informationModel(0)}

q823ObjectClass OBJECT IDENTIFIER ::= {q823-InformationModel managedObjectClass(3)}

q823Package OBJECT IDENTIFIER ::= {q823-InformationModel package(4)}

q823NameBinding OBJECT IDENTIFIER ::= {q823-InformationModel nameBinding(5)}

q823Attribute OBJECT IDENTIFIER ::= {q823-InformationModel attribute(6)}

q823Action OBJECT IDENTIFIER ::= {q823-InformationModel action(7)}

q823Notification OBJECT IDENTIFIER ::= {q823-InformationModel notification(8)}

-- *assignation des valeurs d'extension des aspects liés à l'origine*

```

    extendOrigin OBJECT IDENTIFIER ::= {q823-InformationModel standardSpecificExtension(0)
extendOrigin(0)}

originated Origin ::= originExtension : {extendOrigin 1}

transit Origin ::= originExtension : {extendOrigin 2}

inboundTerminating Origin ::= originExtension : {extendOrigin 3}

-- assignation des valeurs d'extension des aspects liés à la destination

extendDestination OBJECT IDENTIFIER ::= {q823-InformationModel standardSpecificExtension(0)
extendDestination(1)}

hardToReach DestinationAspect ::= destinationExtension : {extendDestination 1}

nonHardToReach DestinationAspect ::= destinationExtension : {extendDestination 2}

-- définition des valeurs par défaut

defaultAdministrativeState AdministrativeState ::= unlocked

defaultAllTmCircuitEndPointSubgroups TmCircuitEndPointSubgroupList ::= {}

defaultAllDestinationCodes DestinationCode ::= ""

defaultDestinationAspect DestinationAspect ::= definedDestinationAspect : null

defaultOriginAspect OriginationAspect ::= {}

defaultRoutingAspects RoutingAspect ::= null

-- définitions des valeurs initiales

initialCount Count ::= 0

initialGauge ObservedValue ::= integer: 0

initialInteger INTEGER ::= 0

-- définitions des valeurs permises

PermittedState ::= AdministrativeState(locked | unlocked)

PermittedHistoryRetention ::= HistoryRetention(1)

PermittedTarFromReturnAction ::= ReturnAction (WITH COMPONENTS {return, cancelTreatment})

-- productions prises en charge

ActivationThresholds ::= SEQUENCE {
    level1 [0] ThresholdLevel,
    level2 [1] ThresholdLevel OPTIONAL}

AdcTriggerType ::= ENUMERATED {
    centralized (0),
    decentralized (1),
    both (2) -- les deux, centralisé et décentralisé --
}

```

```

AdcType ::= ENUMERATED {
    centralized (0),
    decentralized (1)}
AssocAccAffectedTraffic ::= ObjectInstance -- pointe vers une instance accAffectedTraffic

AssocOwnerDccGroup ::= ObjectInstance -- pointe vers une instance dccGroup

AssocScrAffectedTraffic ::= ObjectInstance -- pointe vers une instance scrAffectedTraffic

Category ::= BIT STRING (SIZE(8)) -- valeur conforme à la Recommandation Q.763

CongestionLevel ::= ENUMERATED {
    mcl0 (0),
    mcl1 (1),
    mcl2 (2),
    mcl3 (3)}

ContinuousTimer ::= SEQUENCE {
    calls [0] INTEGER,
    perTimeUnit [1] Timer
}

CreatorIdentity ::= CHOICE {
    sourceCls SourceClass,
    sourceName GraphicString
}

DefinedOrigin ::= ENUMERATED {
    originated (0),
    transit (1),
    inboundTerminating (2)
}

DestinationAspect ::= CHOICE {
    definedDestinationAspect DefinedDestinationAspect,
    destinationExtension OBJECT IDENTIFIER
}

DefinedDestinationAspect ::= ENUMERATED {
    null (0), -- tous les aspects de destination
    htr (1),
    nhtr (2)
}

DestinationCode ::= GraphicString(FROM("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "A" | "B" | "C" | "D" | "E" | "F" | "#" | "*"))

DestinationType ::= CHOICE {
    natureOfAddress [0] NatureOfAddress,
    desttype [1] DestType,
    -- "... Ces points de suspension définis dans un supplément ASN.1 servent ici à indiquer que le type est --
    -- extensible et que des choix supplémentaires pourront être ajoutés dans le futur --
}

DestType ::= ENUMERATED {
    international (0),
    national (1),
    local (2),
    other (3)
}

DispositionOfCall ::= CHOICE {

```

```

skip          NULL,
treatment    Treatment
}

```

```

InstancePointerOrName ::= CHOICE {
  objectInstance [0] ObjectInstance,
  symbolic       [1] NameType
}

```

```

Integer ::= INTEGER(0 .. MAX)

```

```

IntegerValue ::= INTEGER(0 .. 100)

```

```

LeakyBucket ::= SEQUENCE {
  bucketSize [0] INTEGER OPTIONAL, -- l'élément de réseau définira la taille bucketSize
                                         -- si elle n'est pas fournie
  calls      [1] INTEGER,
  perTimeUnit [2] Time (WITH COMPONENTS{scale1, scale2})
}

```

```

NatureOfAddress ::= BIT STRING (SIZE(7)) – valeur conforme à la Recommandation Q.763

```

```

NewTmCircuitEndPointSubgroups ::= SEQUENCE OF InstancePointerOrName

```

```

Origin ::= CHOICE {
  definedOrigin [0] DefinedOrigin,
  namedOrigin   [1] InstancePointerOrName, - -pointe vers une origine ou la nomme.
  originExtension [2] OBJECT IDENTIFIER
}

```

```

OriginationAspect ::= SEQUENCE {
  callingPartyCategory [0] Category OPTIONAL,
  origin [1] Origin OPTIONAL
}

```

```

Percentage ::= CHOICE {
  granularity12p5 [0] ENUMERATED {
    perc0 (0),          --0%--
    perc12p5 (2),      --12,5%--
    perc25 (4),        --25%--
    perc37p5 (6),      --37,5%--
    perc50 (8),        --50%--
    perc62p5 (10),     --67,5%--
    perc75 (12),       --75%--
    perc87p5 (13),     --87,5%--
    perc100 (15) -    --100%--
  },
  granularity10 [1] ENUMERATED {
    perc0 (0) -- 0%--,
    perc10 (1) -- 10%--,
    perc20 (2) -- 20%--,
    perc30 (3) -- 30%--,
    perc40 (4) -- 40%--,
    perc50 (5) -- 50%--,
    perc60 (6) -- 60%--,
    perc70 (7) -- 70%--,
    perc80 (8) -- 80%--,
    perc90 (9) -- 90%--,
    perc100 (10) -- 100%--
  }
}

```

```

ResponseCategories ::= SET OF SEQUENCE {
    originationAspect    [0]    OriginationAspect,
    destinationAspect    [1]    DestinationAspect,
    routingAspect        [2]    RoutingAspect,
    additionalCriteria    [3]    OBJECT IDENTIFIER OPTIONAL,
    percentage           [4]    Percentage
}

ReturnAction ::= CHOICE {
    return                [0]    NULL,
    skip                  [1]    NULL,
    cancelTreatment      [2]    Treatment
}

RoutingAspect ::= ENUMERATED {
    directRoutedTraffic    (0),
    alternateRoutedTraffic (1),
    null                   (2) -- trafic de premier choix ou de débordement
}

SourceClass ::= ENUMERATED {
    tnmos                 (0),
    otherNE               (1),
    thisNE                (2),
    other                 (3)
}

StateIndicatorBitMap ::= BIT STRING {
    exchangeCongestionLevel1    (0),
    exchangeCongestionLevel2    (1),
    congestionLevel1Received    (2),
    congestionLevel2Received    (3),
    scrTriggered                (4),
    accTriggered                (5),
    protectiveControlActive     (6),
    expansiveControlActive      (7),
    destinationControlActive    (8),
    htrDestinationActive        (9),
    circuitEndPointSubgroupAddedOrDeleted(10),
    accTransmissionInhibited    (11),
    adcTriggered                (12)
}

ThresholdLevel ::= CHOICE {
    percentageOfCircuits [0]    INTEGER(0 .. 100),
    noOfCircuits         [1]    INTEGER
}

TmCircuitEndPointSubgroupList ::= SET OF InstancePointerOrName

Treatment ::= CHOICE {
    other                [0]    OBJECT IDENTIFIER,
    announcementNumber  [1]    INTEGER,
    congestionTone       [2]    NULL
}

Timer ::= CHOICE {
    blockAllCalls [0]    NULL,
    scale1        [1]    ENUMERATED {
        sec0 (0)        -- 0 sec --,
        sec0p1 (1)      -- 0,1 sec --,
        sec0p25 (2)     -- 0,25 sec --,
        sec0p5 (3)      -- 0,5 sec --,
        sec1 (4)        -- 1 sec --,
        sec2 (5)        -- 2 sec --,
        sec5 (6)        -- 5 sec --,
    }
}

```

```

sec10 (7)          -- 10 sec --,
sec15 (8)          -- 15 sec --,
sec30 (9)          -- 30 sec --,
sec60 (10)         -- 60 sec --,
sec120 (11)        -- 120 sec --,
sec300 (12)        -- 300 sec --,
sec600 (13)        -- 600 sec--
    },
scale2            [2]  INTEGER(0 .. 600000) -- échelle en milliseconde
}
TrueFalse ::= BOOLEAN
END

```

APPENDICE I

Service de gestion des réseaux de gestion des télécommunications: gestion du trafic

Le présent appendice identifie les fonctions de gestion du trafic qui sont définies dans la Recommandation M.3200 et qui sont prises en charge par la présente Recommandation ainsi que celles qui en dépassent le cadre. Voir le Tableau I.1.

Tableau I.1/Q.823 – Flux d'information Q3

	Elément de réseau	Système d'exploitation
Fonctions de gestion du réseau		Définition dans (Rec.)
1 Fonctions de surveillance du statut		
Rapport sur la disponibilité de service des éléments de réseaux	<-----	M.3100
Rapport, à la demande, sur le statut des commandes, sur la demande	<-----	Q.823
Rapport sur l'état occupé/inactif des faisceaux de circuits	<-----	indirectement
Rapport sur l'état d'encombrement des commutateurs	<-----	Q.823
Rapport sur la réception d'un signal de réduction automatique de l'encombrement	<-----	Q.823
Rapport, à la demande, sur le statut HTR des destinations	<-----	Q.823
Addition/suppression manuelle du statut HTR des destinations	----->	Q.823
Rapport sur le statut d'encombrement du réseau sémaphore	<-----	hors cadre
Rapport sur la réception de signaux de gestion du réseau sémaphore	<-----	hors cadre
2 Fonctions de surveillance des performances		
Fourniture programmée de données et paramètres de faisceaux de circuits	<-----	Q.822, Q.823
Fourniture, à la demande, de données et paramètres de faisceaux de circuits	<-----	Q.822, Q.823
Fourniture programmée de résultats de mesure de la charge des commutateurs	<-----	Q.822, Q.823

Tableau I.1/Q.823 – Flux d'information Q3 (fin)

	Elément de réseau	Système d'exploitation
Fonctions de gestion du réseau		Définition dans (Rec.)
Fourniture, à la demande, de résultats de mesure de la charge de trafic	<-----	Q.822, Q.823
Fourniture programmée de données d'encombrement des commutateurs	<-----	Q.822, Q.823
Fourniture, à la demande, de résultats de mesure de l'encombrement des commutateurs	<-----	Q.822, Q.823
Rapport programmé sur les mesures de charge du réseau sémaphore	<-----	hors cadre
Rapport, à la demande, sur les mesures de charge du réseau sémaphore	<-----	hors cadre
Rapport programmé sur les mesures de l'encombrement du réseau sémaphore	<-----	hors cadre
Rapport, à la demande, sur les mesures de l'encombrement du réseau sémaphore	<-----	hors cadre
Fourniture programmée de données sur le fonctionnement des commandes	<-----	Q.822, Q.823
Rapport, à la demande, de données sur le fonctionnement des commandes	<-----	Q.822, Q.823
3 Fonctions de commande		
Application/modification/suppression d'une commande manuelle	----->	Q.823
Création/modification/suppression d'une commande automatique	----->	Q.823
Activation/désactivation d'une commande automatique	----->	Q.823
Application/modification/suppression d'une annonce enregistrée spéciale	----->	hors cadre
4 Fonctions administratives		
Création/modification/suppression du calendrier de mesure	----->	X.746, Q.823
Création/modification/mise à jour d'une base de données de gestion du réseau	----->	hors cadre
Création/modification/suppression de seuils pour signalisation de statut, fourniture de données et détermination et difficulté HTR	----->	Q.822, Q.823
Création/modification/suppression du calendrier de rapport sur le statut et les données	----->	Q.822, Q.823
Renseignement, à la demande, sur la table d'acheminement		
a) correspondance entre les sous-faisceaux et les destinations	<-----	hors cadre
b) destinations acheminées sur un sous-faisceau de circuits donné	<-----	hors cadre

APPENDICE II

Corrélation entre les commandes trafic définies dans la Recommandation E.412 et les classes d'objets définies dans la présente Recommandation

Le Tableau II.1 donne une corrélation entre les commandes de trafic définies dans la Recommandation E.412 et les classes d'objets définies dans la présente Recommandation.

Tableau II.1/Q.823

Commande de trafic définie dans la Recommandation E.412	Traitement par les classes d'objets définies dans la Recommandation Q.823	Remarques
Code blocking	destinationCodeControl, dccGroup	néant
Call gapping	destinationCodeControl, dccGroup	néant
Cancel direct routing	cancelTo	néant
Cancel alternate routing To	cancelTo	néant
Cancel alternate routing From	cancelFrom	néant
Circuit directionalization	–	hors cadre
Circuit turndown/busying/blocking	–	hors cadre
Skip	skip	néant
Temporary alternate routing on circuit subgroup/destination	tarTo, tarFrom	néant
Cancel rerouted overflow	cancelRerouted	néant
Automatic congestion control	acc, accTrigger, accAffectedTraffic	néant
Selective circuit reservation	scr, scrAffectedTraffic	néant
Automatic destination control	adcTrigger, adc	néant

APPENDICE III

Diagrammes de transition d'état des commandes automatiques et manuelles

Voir les Figures III.1 et III.2.

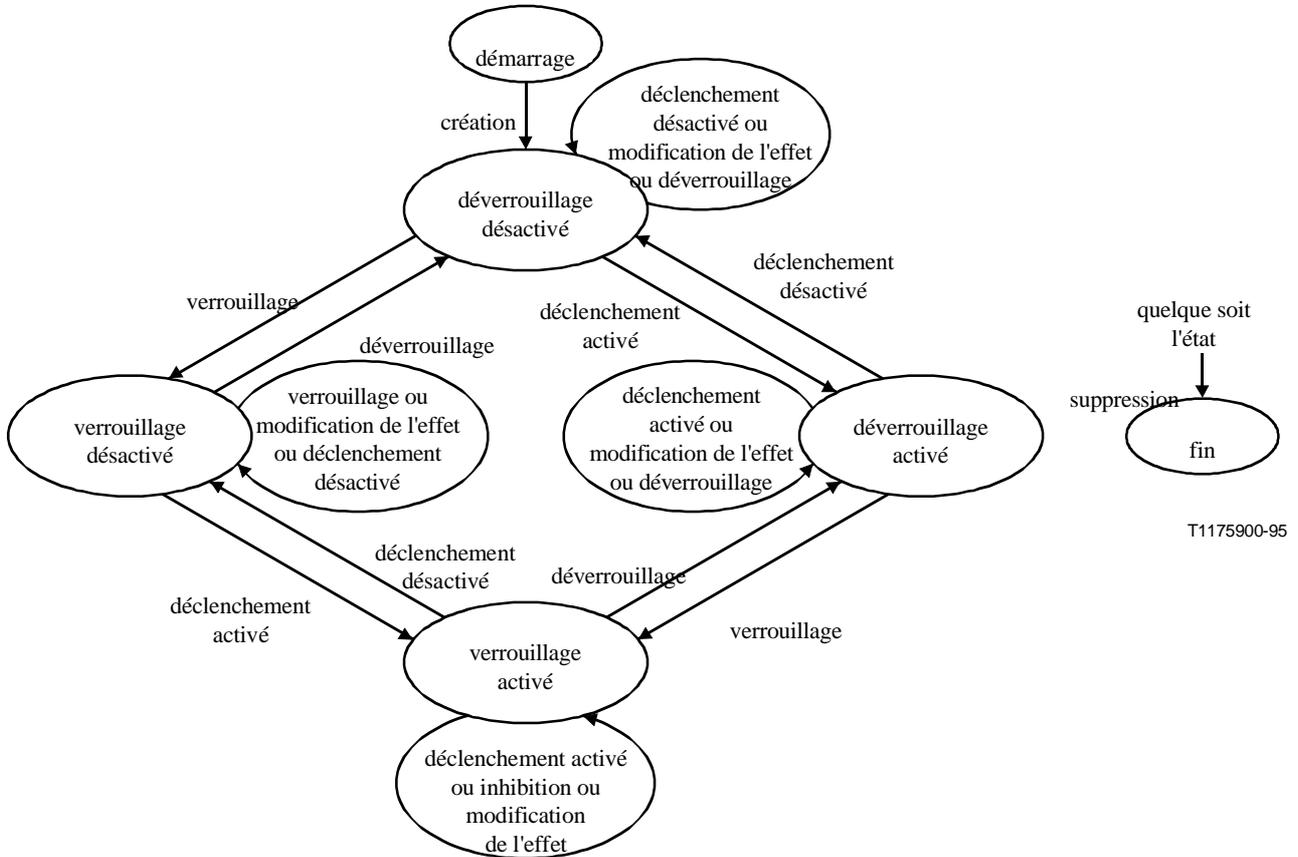


Figure III.1/Q.823 – Diagramme de transition d'état des commandes automatiques

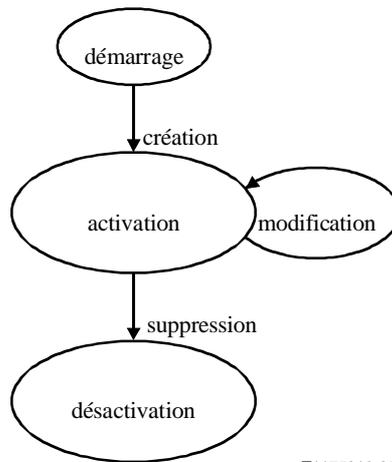


Figure III.2/Q.823 – Diagramme de transition d'état des commandes manuelles

APPENDICE IV

Emploi des commandes d'annulation du débordement vers (*CancelTo*) et d'annulation du débordement depuis (*CancelFrom*) aux trafics de premier choix et de débordement

Afin d'utiliser efficacement les ressources du réseau en période de pointe, le gestionnaire de réseau applique aux sous-faisceaux de circuits les commandes d'annulation du débordement vers et d'annulation du débordement depuis. Ce faisant, il doit avoir la capacité de distinguer et de commander sélectivement les trafics de premier choix et de débordement avec les deux types de commandes. La Figure IV.1 indique les différents types de trafic et explique comment le gestionnaire de réseau peut leur appliquer sélectivement les différentes commandes.

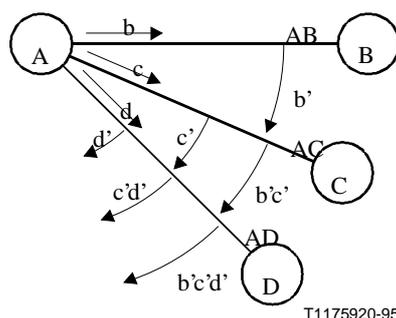


Figure IV.1/Q.823

Dans cette figure, on décrit les types de trafic suivants:

- b est le trafic de premier choix offert au sous-faisceau de circuits AB;
- c est le trafic de premier choix offert au sous-faisceau AC;
- d est le trafic de premier choix offert au sous-faisceau AD;
- b' est un débordement de b reporté de AB sur AC;
- c' est un débordement de c reporté de AC sur AD;
- d' est un débordement de d reporté de AD sur le sous-faisceau de circuits suivant;
- b'c' est un débordement de b' reporté à nouveau de AC sur AD;
- c'd' est un débordement de c' reporté à nouveau de AD sur le sous-faisceau de circuits suivant;
- b'c'd' est un débordement de b'c' reporté à nouveau de AD sur le sous-faisceau de circuits suivant.

A noter que b' et c' n'ont débordé qu'une fois (c'est le trafic de premier choix qui a débordé); tandis que b'c' et c'd' ont débordé plus d'une fois (c'est le trafic de débordement qui a débordé).

Le gestionnaire du réseau doit disposer de la flexibilité nécessaire pour commander chacun de ces types de trafic avec les commandes d'annulation du débordement vers et d'annulation du débordement depuis, et pour distinguer plus particulièrement le trafic qui n'a débordé qu'une fois de celui qui a débordé plus d'une fois. Le Tableau IV.1 indique comment on pourrait procéder:

Tableau IV.1/Q.823

Trafic à commander	Type de commande
b	annulation du débordement du trafic de premier choix vers AB
c	annulation du débordement du trafic de premier choix vers AC
d	annulation du débordement du trafic de premier choix vers AD
b'	annulation du débordement du trafic de premier choix depuis AB
c'	annulation du débordement du trafic de premier choix depuis AC
d'	annulation du débordement du trafic de premier choix depuis AD
b'c'	annulation du débordement du trafic de débordement depuis AC
c'd'	annulation du débordement du trafic de débordement depuis AD
b'c'd'	annulation du débordement du trafic de débordement depuis AD

Il est à noter que si les sous-faisceaux de circuits AB, AC et AD font aussi partie des autres tables d'acheminement, leurs débordements les affecteront aussi.

APPENDICE V

Utilisation de la classe de données chronologiques, *historyData*, définie dans la Recommandation Q.822 pour les mesures du trafic

V.1 Introduction

Le présent appendice fournit un guide d'instructions sur l'application du modèle d'information de gestion des performances, qui est défini dans la Recommandation Q.822 et qui concerne la collecte des mesures de trafic. Ce guide fournit des mécanismes de rapports émis de façon autonome et d'extraction de mesures à la demande.

V.2 Aperçu général du modèle d'information de gestion des performances

Le modèle de gestion des performances active les rapports récapitulatifs portant sur les paramètres de qualité au moyen du releveur *simpleScanner*.

Son modèle d'objet est montré sur la Figure 1/Q.822.

Les données trouvées dans les instances de la classe d'objets *currentData* (ou de ses sous-classes) comportent des mesures qui figurent dans le processus d'accumulation et de changement effectué durant l'intervalle de relève. A la fin de cet intervalle, les instances d'objets *currentData* sont initialisées. Les données collectées doivent donc être immédiatement rapportées ou mémorisées de façon à garder en mémoire les informations relatives à ces intervalles terminés. Le modèle prend en charge la conservation du dernier intervalle complet sous forme d'instance d'objet *historyData* (ou de ses sous-classes).

A la fin de chaque intervalle de relève des performances, un rapport *scanReport* peut être émis directement par l'objet *currentData*. Toutefois, cette opération donne lieu à une notification distincte pour chaque instance *currentData* à surveiller.

Il est plus efficace de synthétiser les mesures de toutes les instances *currentData* accumulées durant la même période. Le mode qu'il est recommandé d'utiliser pour synchroniser les instances *historyData* est le mode *simpleScanner*. L'objet géré *simpleScanner* relève les attributs communs à observer dans chacun des objets sélectionnés en fonction d'un calendrier spécifié et produit un rapport récapitulatif.

Les notifications générées par le releveur peuvent être traitées par un discriminateur de retransmission d'événement (EFD, *event forwarding discriminator*) pour être rapportées au système d'exploitation.

La génération des rapports récapitulatifs de gestion des performances est illustrée sur la Figure B.5/Q.822. Cette illustration montre que le releveur *simpleScanner* peut servir à relever les attributs des objets *historyData* et émettre un rapport *scanReport* au discriminateur. Celui-ci à son tour transmet un rapport récapitulatif m-EVENT assurant le respect des assertions de valeurs établies (*eventType = scanReport*, par exemple).

V.3 Configuration initiale

Ce sous-paragraphe discute le processus de configuration du mécanisme de collecte.

Les classes d'objets suivantes sont instanciées:

- 1) sous-classes *currentData*: *tmCircuitEndPointSubgroupCurrentData*, par exemple;
- 2) sous-classes *historyData*: *xtpsgHistoryData*, par exemple;
- 3) *simpleScanner*.

V.3.1 Données courantes

L'intervalle de collecte est spécifié dans l'attribut *granularityPeriod* qui est hérité du releveur. La classe *currentData* est une sous-classe de la classe *scanner*. Les valeurs types sont par exemple cinq minutes ou soixante secondes.

Le progiciel conditionnel *historyRetentionPkg* est requis de façon à établir le nombre approprié d'intervalle mémorisé dans la classe *historyData*. Dans le cadre de la gestion du trafic, cette valeur est limitée à 1 afin de minimiser l'utilisation de ressources de type mémoire.

V.3.2 Données chronologiques

La période (durée) de conservation est spécifiée dans l'attribut *historyRetention*. Lors de la création d'une nouvelle instance *historyData*, il est possible de supprimer l'instance existante la plus ancienne.

Le progiciel conditionnel *historyDataSuspectIntervalPkg* est requis puisqu'il sert à savoir si les mesures sont suspectes.

V.3.3 Releveur simple

Il est recommandé de positionner comme suit les attributs suivants qui sont hérités des releveurs simple ou homogène *homogeneousScanner* afin de réaliser le COMPORTEMENT (BEHAVIOUR) souhaité.

- 1) *scannerId* – identificateur de relève (hérité de la classe *scanner* définie dans la Recommandation X.739)
L'attribut *scannerId* contient une valeur servant à identifier une instance de cette classe d'objets gérés (sert pour la dénomination).
- 2) *administrativeState* – état administratif (hérité de la classe *scanner* définie dans la Recommandation X.739)
L'attribut *administrativeState* (défini dans la Recommandation X.731) sert à suspendre ou à reprendre la relève. Si sa valeur est UNLOCKED, le releveur peut effectuer des relèves. Sa valeur initiale doit être UNLOCKED plutôt que LOCKED de façon à permettre la relève.
- 3) *granularityPeriod* – période granulaire (hérité de la classe *scanner* définie dans la Recommandation X.739)

L'attribut *granularityPeriod* indique le temps écoulé entre des relèves successives. Il doit être réglé sur 5 minutes pour obtenir une relève et un rapport émis de façon autonome à la fin de chaque période.

- 4) *scope* – domaine (hérité de la classe *homogeneousScanner* décrite dans la Recommandation X.738)

Le progiciel de sélection du domaine est spécifié. Le releveur utilise la classe d'objets gérés identifiée par l'attribut *baseManagedObject* et vérifie tous les objets gérés implantés aux niveaux indiqués par l'attribut *scope*. L'objet *baseManagedObject*, qui doit être le nœud de départ pour le choix de l'objet géré, est l'élément *managedElement* (le commutateur de circuits par exemple). La valeur de l'attribut *scope* sera par exemple positionnée sur le niveau 2. Ce choix implique que tous les objets du deuxième niveau de l'arborescence de dénomination relatifs à l'élément *managedElement* sont candidats à la sélection. Celle-ci est limitée plus tard par le filtre de relève (*scanningFilter*) comme cela est expliqué ci-dessous.

- 5) *scanningFilter* – filtre de relève (hérité de la classe *homogeneousScanner* décrite dans la Recommandation X.738)

Le releveur vérifie les instances d'objets gérés implantées aux niveaux indiqués en appliquant le critère choisi dans l'attribut *scanningFilter*.

Le filtre peut être configuré de façon à choisir entre les types de données suivants:

- a) sous-faisceaux de circuits;
- b) destination;
- c) commutateur.

Le filtre peut par exemple être configuré pour sélectionner les objets de données chronologiques des sous-faisceaux de circuits des entités surveillées, de destination et de commutateur (objectClass = tmCircuitEndPointSubgroupHistoryData OU objectClass = destinationHistoryData OU objectClass = exchangePerformanceHistoryData).

- 6) *numericAttributeIdArray* – matrice numérique des identificateurs *d'attributs* (hérité de la classe *simpleScanner* décrit dans la Recommandation X.738)

L'attribut *numericAttributeIdArray* doit être valué. Il s'agit d'une liste ordonnée des identificateurs d'attributs dont les valeurs doivent être relevées. Celles-ci seront contenues dans le tableau *numericValueArray*.

- 7) *supressObjectInstance* – suppression de l'identification de l'instance d'objet (hérité de la classe *simpleScanner* décrit dans la Recommandation X.738)

L'attribut *supressObjectInstance* est positionné sur Vrai de façon à ne pas inclure l'identification de l'instance d'objet dans la réponse. Cette procédure réduira le nombre d'octets du message.

NOTE – On doit éviter ce positionnement si aucun progiciel de suppression de la classe *currentData* n'est inclus.

V.4 Rapport émis de façon autonome

Le mécanisme d'émission autonome des rapports est illustré sur la Figure V.1 intitulée Rapport de mesures émis de façon autonome.

Il est possible d'invoquer le releveur de façon autonome au début de chaque période granulaire. La génération autonome des rapports intervient périodiquement en positionnant l'attribut *granularityPeriod* du releveur *simpleScanner* sur une valeur appropriée: 5 minutes, par exemple. Un relevé des données chronologiques appropriées à chaque objet surveillé sera effectué. Les valeurs pertinentes des attributs seront insérées dans la matrice *numericValueArray* de la NOTIFICATION du rapport *scanReport*.

Cette matrice peut comporter une séquence de nombre entiers ou réels. L'ordre exact des identificateurs d'attributs est connu à partir de l'identificateur *numericAttributeIdArray*.

La notification est envoyée au discriminateur de retransmission d'événement.

La NOTIFICATION *scanReport* est évaluée par le discriminateur. Celui-ci doit donc être construit de façon à pouvoir envoyer les notifications appropriées. Les rapports récapitulatifs seront envoyés au gestionnaire sous forme des rapports m-EVENT.

V.5 Rapport à la demande

Le mécanisme de rapport à la demande est illustré sur la Figure V.2 intitulée Rapport de mesures émis sur demande.

Un mécanisme à la demande peut être réalisé en mettant l'attribut *granularityPeriod* du releveur *simpleScanner* à zéro. Cette opération désactive le mode d'émission autonome. Le gestionnaire (système d'exploitation) peut lancer une relève à la demande en invoquant l'ACTION *activateScanReport* du releveur simple. Avec cette procédure, le releveur accomplit sa tâche en une seule fois (collecte de données et réponses avec résultats correspondants). L'action sera exécutée en affectant aux attributs *scope* et *scanningFilter* les valeurs des attributs qui servent lors de l'émission autonome des rapports.

Le rapport récapitulatif sera envoyé au gestionnaire sous forme d'une réponse m-ACTION.

V.6 Changement du rapport

Le gestionnaire peut suspendre la génération autonome des rapports du releveur *simpleScanner* en positionnant l'attribut *administrativeState* sur verrouillé. Ce positionnement va interrompre brusquement l'utilisation de la relève. Il est possible d'avoir une approche plus élégante en positionnant l'attribut *administrativeState* sur arrêt. La génération autonome s'arrêtera à la fin de l'intervalle courant et après l'émission du rapport.

Le gestionnaire peut changer les valeurs des attributs *scope*, *scanningFilter* et *numericAttributeIdArray* pour intervenir sur le contenu du rapport. Il est aussi possible de changer d'autres paramètres afin de modifier le comportement du mécanisme de rapport: la période *granularityPeriod*, par exemple.

V.7 Conclusions

Les mécanismes normalisés, qui spécifient la gestion des performances définie dans la Recommandation Q.822, peuvent servir conjointement avec le releveur *simpleScanner* décrit dans la Recommandation X.738 pour récapituler et rapporter les mesures de trafic. Les rapports émis sur demande et de façon autonome peuvent être pris en charge en utilisant les deux mécanismes décrits.

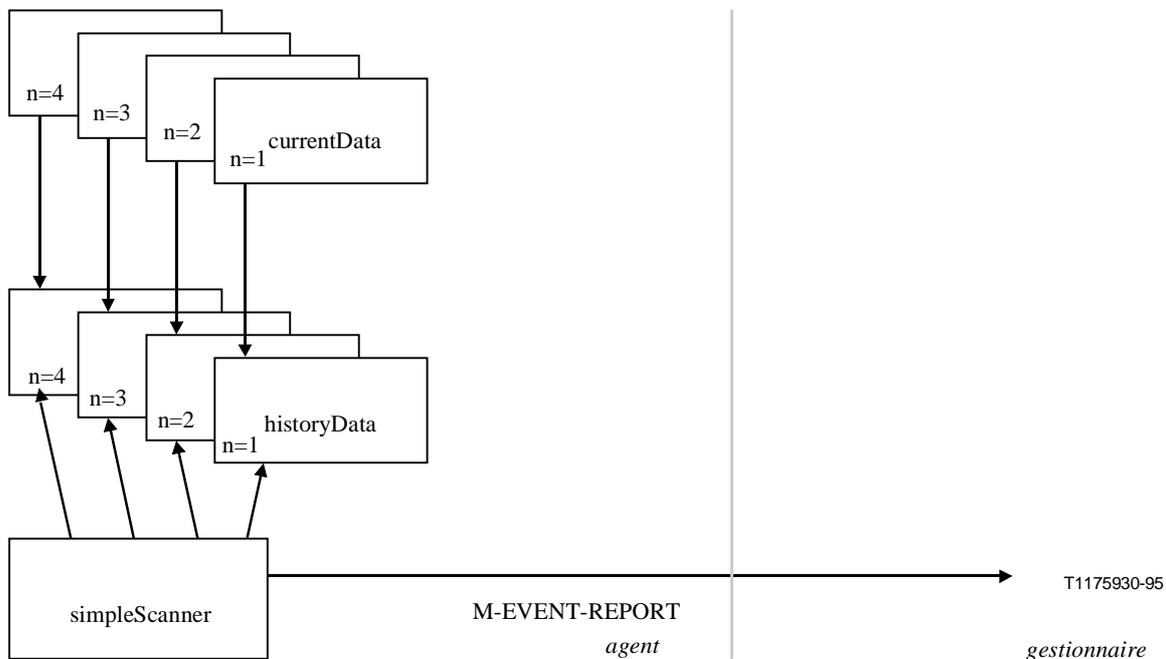


Figure V.1/Q.823 – Rapport de mesures émis de façon autonome

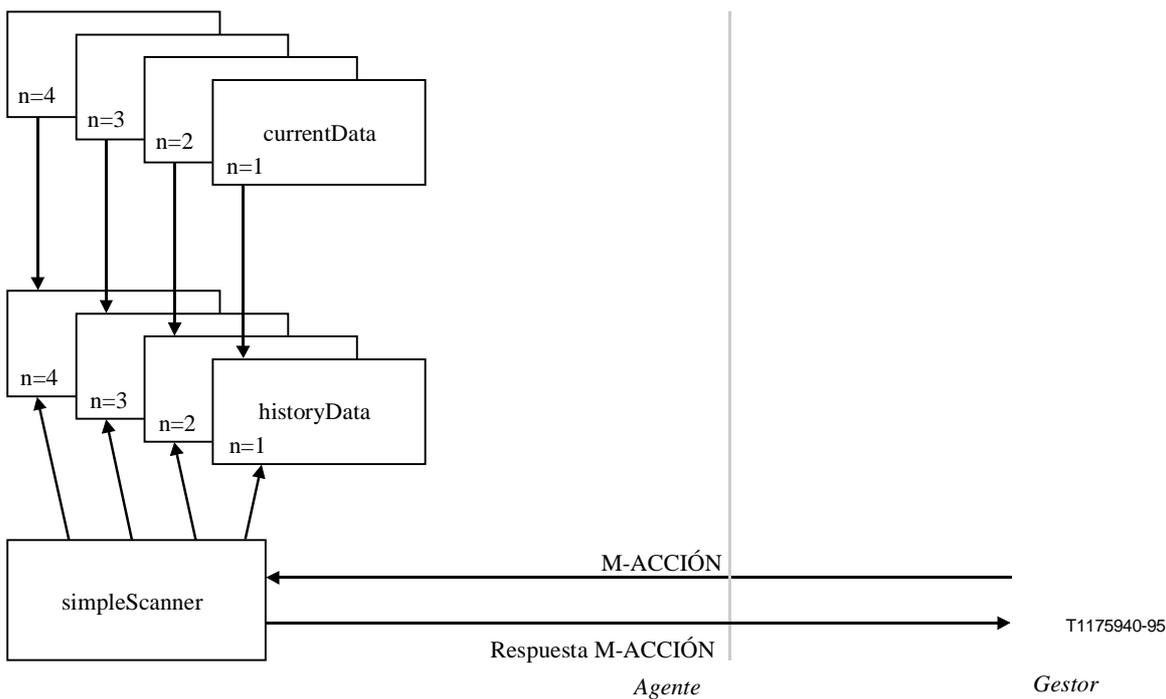


Figure V.2/Q.823 – Rapport de mesures émis sur demande

APPENDICE VI

Définition de l'objet *CircuitEndPointSubgroup* (sous-faisceau d'extrémité de circuits)

En attendant de déplacer dans la Recommandation M.3100 la définition de l'objet *circuitEndPointSubgroup* suivante, celle-ci est incluse dans le présent Appendice dans un souci de compréhension et d'exhaustivité du modèle Q.823.

```
circuitEndPointSubgroup      MANAGED OBJECT CLASS
DERIVED FROM "Recommendation X.721 :1992":top;
CHARACTERIZED BY
  circuitEndPointSubgroupPackage PACKAGE
  BEHAVIOUR
    circuitEndPointSubgroupBehaviour BEHAVIOUR
DEFINED AS
  "Ensemble d'extrémités de circuits d'un faisceau qui interconnecte directement deux commutateurs dont
  les valeurs des attributs listés dans le progiciel sont communes. Il est à noter que le terme commutateur
  inclut les autocommutateurs PBX lorsque ce cas est applicable. – L'Annexe A/E.410 définit un sous-faisceau
  de circuits.
  ;;
ATTRIBUTES
  circuitEndPointSubgroupId          GET,
  numberOfCircuits                  GET,
  labelOfFarEndExchange             GET,
  signallingCapabilities           GET,
  informationTransferCapabilities   GET,
  circuitDirectionality            GET,
  transmissionCharacteristics      GET,
  userLabel                         GET;
  ;;
NOTIFICATIONS
  "Recommendation X.721:1992":attributeValueChange,
  "Recommendation X.721:1992":objectCreation,
  "Recommendation X.721:1992":objectDeletion,
REGISTERED AS { m3100ObjectClass x};
circuitEndPointSubgroupId ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
  MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
  BEHAVIOUR
    "Recommendation X.721 :1992": rDNIdBehaviour,
    -- Le comportement ci-dessus est défini dans le cadre de l'identificateur de discriminateur
    -- discriminatorId décrit dans la Recommandation X.721
    ttpIdBehaviour BEHAVIOUR
  DEFINED AS "L'identificateur de sous-faisceau de circuits est un type d'attribut dont la valeur distinctive
  peut servir comme nom RDN quand on nomme une instance de la classe d'objets du sous-faisceau de circuits.";;
REGISTERED AS {m3100Attribute x};
numberOfCircuits ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NumberOfCircuits;
  MATCHES FOR EQUALITY, ORDERING;
  DEFINED AS "Nombre de circuits d'un sous-faisceau.";;
REGISTERED AS {m3100Attribute x};
labelOfFarEndExchange ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.UserLabel;
  MATCHES FOR EQUALITY, SUBSTRINGS;
  BEHAVIOUR
```

labelOfFarEndExchangeBehaviour BEHAVIOUR
DEFINED AS "Ce type d'attribut assigne un nom convivial au commutateur distant qui termine le sous-faisceau de circuits.";;

REGISTERED AS {m3100Attribute x};

signallingCapabilities ATTRIBUTE
WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.SignallingCapabilities;
MATCHES FOR EQUALITY;
BEHAVIOUR
signallingCapabilitiesBehaviour BEHAVIOUR
DEFINED AS "Ce type d'attribut spécifie les types de signalisation pris en charge par le sous-faisceau de circuits.";;

REGISTERED AS {m3100Attribute x};

informationTransferCapabilites ATTRIBUTE
WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.InformationTransferCapabilities;
MATCHES FOR EQUALITY;
BEHAVIOUR
informationTransferCapabilitesBehaviour BEHAVIOUR
DEFINED AS "Ce type d'attribut spécifie les différents types de service pris en charge par le sous-faisceau de circuits: la parole, toutes les données à 64 kbit/s, par exemple.";;

REGISTERED AS {m3100Attribute x};

circuitDirectionality ATTRIBUTE
WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.CircuitDirectionality;
MATCHES FOR EQUALITY;
BEHAVIOUR
circuitDirectionalityBehaviour BEHAVIOUR
DEFINED AS "Ce type d'attribut spécifie la directionnalité des circuits du sous-faisceau";;

REGISTERED AS {m3100Attribute x};

transmissionCharacteristics ATTRIBUTE
WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.TransmissionCharacteristics;
MATCHES FOR EQUALITY;
BEHAVIOUR
transmissionCharacteristicsBehaviour BEHAVIOUR
DEFINED AS "Ce type d'attribut spécifie les différentes caractéristiques de transmission du sous-faisceau de circuits: liaison satellitaire, existence de limitation d'écho. Les bits sont positionnés afin d'indiquer si une caractéristique particulière est prise en charge.";;

REGISTERED AS {m3100Attribute x};

ASN.1 productions:

NumberOfCircuits ::= INTEGER

SignallingCapabilities ::= ENUMERATED {isup(0), isup2(1),ccittNo5(2),r2(3),ccittNo6(4), tup(5)
-- "... Ces points de suspension définis dans un supplément ASN.1 servent ici à indiquer que ce –
-- type est extensible et que d'autres valeurs de type énuméré pourront être ajoutées dans le futur --}

CircuitDirectionality ::= ENUMERATED {onewayOut(0), onewayIn(1), twoway(2)}

InformationTransferCapabilites ::= ENUMERATED {speech(0),
audio3pt1(1),audio7(2),audioComb(3),digitalRestricted56(4), digitalUnrestricted64(5)
-- "... Ces points de suspension définis dans un supplément ASN.1 servent ici à indiquer que ce –
-- type est extensible et que d'autres valeurs de type énuméré pourront être ajoutées dans le futur --}

TransmissionCharacteristics ::= BITSTRING {satellite(0), dCME(1), echoControl(2)}
 -- La création et la suppression d'un objet *CircuitEndPointSubgroup* avec la corrélation de nom suivante dépasse le cadre de la Recommandation Q.823.

circuitEndPointSubgroup-managedElement NAME BINDING
SUBORDINATE OBJECT CLASS circuitEndPointSubgroup;
NAMED BY
SUPERIOR OBJECT CLASS "M.3100":managedElement AND SUBCLASSES;
WITH ATTRIBUTE circuitEndPointSubgroupId;
CREATE WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE;

APPENDICE VII

Mappage des fonctions de gestion du réseau sur les unités fonctionnelles

Le Tableau VII.1 suivant fournit un mappage des fonctions de gestion du réseau (énumérées dans l'Appendice I) sur les unités fonctionnelles.

Tableau VII.1/Q.823

Fonctions de gestion du réseau	Unités fonctionnelles (paragraphe 8)
1 Fonctions de surveillance du statut	
Rapport sur la disponibilité de service des éléments NE	H
Rapport sur le statut des commandes, sur demande	H, I ou J
Rapport sur l'état occupé/inactif des faisceaux de circuits	indirectement – voir fourniture à la demande de données et de paramètres des faisceaux de circuits
Rapport sur l'état d'encombrement des commutateurs	H, I ou J
Rapport sur la réception d'un signal de réduction automatique de l'encombrement	I ou J
Rapport, sur demande, sur le statut des destinations difficiles à atteindre	G, H
Addition/suppression manuelle du statut HTR des destinations	G
Rapport sur le statut d'encombrement du réseau sémaphore	hors cadre
Rapport sur la réception des signaux de gestion du réseau sémaphore	hors cadre
2 Fonctions de surveillance des performances	
Fourniture programmée de données et paramètres de faisceaux de circuits	H, D, ii)
Fourniture, à la demande, de données et paramètres de faisceaux de circuits	H, D, I
Fourniture programmée de résultats de mesure de la charge des commutateurs	C, ii)

Tableau VII.1/Q.823 (fin)

Fonctions de gestion du réseau	Unités fonctionnelles (paragraphe 8)
Fourniture, à la demande, de résultats de mesure de la charge de trafic	C, I
Fourniture programmée de données d'encombrement des commutateurs	C, ii)
Fourniture, à la demande, de résultats de mesure de l'encombrement des commutateurs	C, I
Rapport programmé sur les mesures de la charge du réseau sémaphore	hors cadre
Rapport, à la demande, sur les mesures de la charge du réseau sémaphore	hors cadre
Rapport programmé sur les mesures d'encombrement du réseau sémaphore	hors cadre
Rapport, à la demande, sur les mesures d'encombrement du réseau sémaphore	hors cadre
Fourniture programmée de données sur le fonctionnement des commandes	B, ii)
Fourniture, à la demande, de données sur le fonctionnement des commandes	B, i)
3 Fonctions de commandes	
Application/modification/suppression d'une commande manuelle	A
Création/modification/suppression d'une commande automatique	A
Application/modification/suppression d'une annonce enregistrée spéciale	hors cadre
4 Fonctions administratives	
Création/modification/suppression du calendrier de mesure	K
Création/modification/mise à jour d'une base de données de gestion du réseau	hors cadre
Création/modification/suppression de seuils pour signalisation de statut, fourniture de données et détermination	ii)
Création/modification/suppression du calendrier de rapport sur le statut et les données	ii)
Renseignement sur la table d'acheminement, à la demande	hors cadre

APPENDICE VIII

Prise en charge de la fonctionnalité de vérification avec le modèle décrit dans la Recommandation Q.823

VIII.1 Introduction

Le présent appendice décrit l'utilisation du modèle fourni dans la présente Recommandation, en le combinant au modèle de la fonction de gestion de la connaissance de la gestion (décrite dans la Recommandation X.750), afin de prendre en charge les capacités de vérification. Les prescriptions fonctionnelles en matière de vérification effectuée par le gestionnaire sont discutées au VIII.2. Les mécanismes susceptibles d'être utilisés pour satisfaire à ces prescriptions sont décrits au VIII.3.

VIII.2 Prescriptions fonctionnelles

La vérification est nécessaire pour que le gestionnaire et l'agent disposent de la même vue sur les données de référence (données non mesurées). Le gestionnaire demande une vérification des données présentes dans la base d'informations de gestion (MIB, *management information base*) de l'élément de réseau (elles sont différentes des données de surveillance de gestion rapportées périodiquement) afin d'aligner les connaissances de sa base MIB sur l'état courant de l'agent. Il lance cette vérification pour une ou plusieurs des raisons suivantes:

- a) suite à des anomalies trouvées dans les données mesurées qui ont été reçues (telles que l'absence de données de mesure relatives au faisceau de jonction bien que l'agent en ait programmé l'envoi);
- b) suite à la création, à la suppression et aux changements effectués sur les valeurs d'informations de gestion;
- c) suite à une alerte que l'agent fournit de façon autonome (classe d'objet d'indicateur d'état *stateIndicator*) et qui concerne par exemple l'encombrement ou les activités de commande,
- d) afin de prendre en charge l'alignement périodique des connaissances de gestion qui sont partagées entre le gestionnaire et l'agent.

Le mécanisme d'extraction des données présentes dans l'agent prendra en charge la personnalisation par le gestionnaire afin de satisfaire un ou plusieurs des critères suivants: type et volume de données ainsi que fréquence d'extraction. Cette extraction prendra en charge les vérifications de toutes les données de référence (vérification complète) et aussi l'extraction sélective de données de référence modifiées depuis la précédente vérification (vérification des modifications).

Ces prescriptions, excepté la vérification des modifications, peuvent être respectées en utilisant les mécanismes décrits ci-dessous. Une approche possible de prise en charge de la vérification des modifications, si elle est requise par le gestionnaire, est décrite plus loin.

VIII.3 Types de vérification

VIII.3.1 Vérification complète ou partielle

La vérification complète extrait toutes les informations de la base MIB de l'élément de réseau pour la classe d'objet sélectionnée. La vérification partielle extrait un ensemble de ces objets en appliquant un filtre à leurs attributs. Les différents mécanismes de demande d'informations de vérification sont décrits dans le VIII.4.

La vérification est demandée par le système d'exploitation:

- a) quand il veut construire la vue initiale de la base MIB de l'élément de réseau;
- b) quand il veut aligner la vue de sa base MIB sur celle de l'élément de réseau.

Dès réception de l'information de vérification, il appartient au système d'exploitation de construire la base MIB initiale ou de déterminer et de corriger les différences incrémentales.

VIII.3.2 Vérification des modifications

En plus de la vérification complète ou partielle, le système d'exploitation peut aussi ne souhaiter recevoir d'information concernant cette vérification que lorsque des changements ont été effectués sur la base MIB de l'élément de réseau depuis la dernière modification. Ces prescriptions peuvent être suivies grâce à l'utilisation de notifications existantes, comme la création, la suppression ou la modification de valeurs d'attributs, qui sont émises à partir d'objets de données de référence. Ces informations peuvent être saisies par le système d'exploitation pour déterminer et corriger les différences incrémentales.

Si une réelle modification est souhaitée, alors les notifications de création, de suppression et de modification de valeurs d'attributs doivent être enregistrées. Le releveur *simpleScanner* peut alors servir à extraire l'information des enregistrements en spécifiant le progiciel de sélection *timeSelection* dans le releveur.

VIII.4 Mécanismes

VIII.4.1 Utilisation du service PT-GET

Le gestionnaire peut lancer une demande en combinant le service PT-GET défini dans la Recommandation X.730 et les caractéristiques de délimitation du domaine et de filtrage disponibles dans l'élément de service commun de transfert des informations de gestion (CMISE) (Recommandations X.710 et X.711). Afin de mettre en œuvre cette capacité, il est nécessaire de prendre en charge les fonctions de profil ISO ISP 11183-2 intitulé "communications étendues d'informations de gestion". Cette partie du profil de gestion du réseau rend obligatoires les unités fonctionnelles optionnelles (notamment sélection d'objets gérés, réponse multiple, filtre et *cancel-GET*). De plus, au moment de l'établissement de l'association, ces unités du service commun de transfert des informations de gestion (CMIS) doivent être négociées pour pouvoir être utilisées sur cette association.

Avec ce mécanisme, le gestionnaire peut émettre une seule demande pour extraire des données correspondant à différents critères. On donne ci-dessous des exemples de conditions (non exhaustives) que le gestionnaire peut demander à l'agent d'appliquer, afin de déterminer si une réponse doit être émise. Une réponse distincte sera envoyée par l'agent pour chaque objet géré qui satisfait aux conditions précisées dans la demande. A tout moment durant l'envoi de plusieurs réponses, le gestionnaire peut annuler la demande en émettant une demande *cancel-GET*.

Le paramètre *CMISfilter* est défini en termes d'assertions portant sur des valeurs d'attributs. Il s'agit d'une expression booléenne évaluée à vrai ou faux. Pour ce qui est du modèle de la présente Recommandation, les assertions peuvent servir à extraire des données appartenant à des objets gérés de commandes ou à des objets de données de référence existants. En utilisant cette capacité, on obtient les valeurs des attributs de ces objets gérés.

A Vérification de commandes

Le paramètre *CMISFilter* assure une extraction sélective en fonction du type de commande. Il est à noter qu'il est possible de combiner plusieurs des conditions de sélection si le gestionnaire doit extraire l'information correspondant à plusieurs types de commande.

- a) Espacement d'appels {objectClass = destinationCodeControl}
- b) Annulation des trafics de premier choix et de débordement vers {objectClass = cancelTo}

- | | | |
|----|--|--|
| c) | Annulation du trafic de débordement depuis | {objectClass = cancelFrom} |
| d) | Evitement | {objectClass = skip} |
| e) | Débordement temporaire vers | {objectClass = tarTo} |
| f) | Débordement temporaire depuis | {objectClass = tarFrom} |
| g) | Annulation du trafic de débordement réacheminé | {objectClass = cancelRerouted} |
| h) | Réduction automatique de l'encombrement | {objectClass = acc} |
| i) | Réservation sélective de circuits | {objectClass = scr} |
| j) | Commande automatique de destination | {objectClass = adc} |
| k) | Toutes les commandes protectrices manuelles | {objectClass = cancelFrom OU
objectClass = cancelTo OU objectClass
= skip} |
| l) | Commandes expansives | {objectClass = tarTo OU objectClass =
tarFrom} |
| m) | Toutes les commandes | {present trafficControlId} |
| n) | Toutes les commandes automatiques actives | {present trafficControlId ET
administrativeState = unlocked ET
autoActivated = true} |

B Données de référence

Les types d'assertions suivants peuvent servir à extraire des données classées comme données de référence dans le modèle.

Il est à noter que cette liste n'est pas exhaustive, mais qu'elle est simplement fournie à titre d'exemple.

- | | | |
|----|--|--|
| a) | Sous-faisceaux de circuits | {objectClass = circuitSubGroupEndPoint} |
| b) | Sous-faisceaux de circuits surveillés | {objectClass=circuitSubGroupEndPoint ET
tmSurveillance = true} |
| c) | Sous-faisceaux de circuits avec une
extrémité distante spécifique | {objectClass=circuitSubGroupEndPoint ET
labelOfFarEndExchange = <value of the specific
exchange >} |
| d) | Destination HTR | {objectClass = htrDestination} |
| e) | Destination observée | {objectClass = observedDestination} |

Ce mécanisme peut servir à effectuer une vérification complète.

VIII.4.2 Utilisation du releveur simple

Le mécanisme décrit ci-dessus résulte d'une seule demande, suivie par un nombre de réponses correspondant à chaque objet sélectionné. Selon les conditions spécifiées, le nombre de réponses liées à la demande peut être très important. Afin de le réduire, il est possible d'utiliser un autre mécanisme, qui fournira cette information en regroupant les réponses concernant plusieurs objets en un seul rapport. L'Appendice V décrit la façon d'utiliser le releveur *simpleScanner* pour extraire les données appartenant aux objets de la classe *historyData*. La capacité de vérification peut être prise en charge en utilisant le même mécanisme. Dans ce cas, les objets gérés soumis à la modification sont obtenus en configurant le progiciel de sélection du domaine avec les conditions spécifiées dans

le mécanisme précédent, conformément à la description figurant à l'Appendice V. Ces informations seront délivrées à la demande (par opposition à une émission autonome de rapport).

Il est à noter que bien que ce mécanisme soit très utile pour grouper les réponses de sélection du domaine en exécutant une vérification complète, certaines mises en œuvre peuvent ne pas être prises en charge lorsqu'il est envoyé et reçu plus de 10 000 octets d'unité de données de protocole de session (les données qui codent l'application devront occuper moins de quelques octets pour que l'information d'en-tête des couches Session et Présentation soit prise en compte). Ce scénario est possible parce que le profil ISP 11183-1 nécessite une mise en œuvre pour prendre en charge l'envoi et à la réception d'au minimum 10 kilo-octets de données. Dans ce cas, la réponse peut être fournie dans plusieurs réponses liées.

VIII.4.3 Utilisation de l'objet de découverte *discoveryObject*

Le premier mécanisme décrit dans le VIII.3.1 requiert que l'agent prenne en charge les caractéristiques de délimitation du domaine et de filtrage de l'élément CMISE sur l'interface. Avec le mécanisme décrit dans le VIII.3.2, même s'il n'est pas nécessaire que le gestionnaire envoie ces caractéristiques dans la demande, ces capacités feront partie du releveur simple de l'agent. La Recommandation X.750 définit une classe d'objets gérés appelée *discoveryObject*. Elle prend en charge un gestionnaire pour obtenir la connaissance de gestion quand l'agent ne prend pas en charge les caractéristiques de délimitation du domaine et de filtrage.

La définition *discoveryObject* donnée dans la Recommandation X.750 comporte une action *mITSearch*. En utilisant cette action, le gestionnaire obtient les instances de différents objets gérés. Il peut aussi optionnellement demander l'attribut *objectClass* correspondant à ces instances. Cette approche est utile si le gestionnaire ne requiert que les noms d'instances et leur classe. Toutefois, si les valeurs des autres attributs sont aussi demandées pour la vérification, il est nécessaire qu'il fasse appel plusieurs fois au service PT-GET.

Le gestionnaire peut émettre des demandes m-Get séparées pour extraire les données d'un ensemble d'instances comprises dans la réponse de l'objet *discoveryObject* ou de toutes les instances qui y figurent. Autrement, il peut créer un objet *simpleScanner* inclus dans la liste d'objets gérés à partir desquels les données doivent être extraites conjointement avec les attributs dont on doit inclure les valeurs dans le rapport.

VIII.5 Conclusions

Le présent Appendice décrit la manière dont le modèle existant prend en charge les capacités de vérification. En combinant les caractéristiques de l'élément CMISE et des modèles d'informations disponibles dans les Recommandations de l'UIT-T, les vérifications des données de références sont prises en charge.

APPENDICE IX

Exemples d'erreurs d'adressage dans la Recommandation Q.823

IX.1 Introduction

En mettant en œuvre une interface Q3 avec les prescriptions de protocole figurant dans les Recommandations Q.811 et Q.812 ainsi qu'un modèle d'information spécifique à une application comme celui de la Recommandation Q.823, des erreurs peuvent apparaître à différents niveaux. Elles peuvent être classées comme:

- des erreurs générées par la machine protocolaire dans différentes couches comportant les éléments ACSE (élément de service de contrôle d'association) et CMISE (élément de service commun d'information de gestion);
- des erreurs génériques définies dans l'élément CMISE mais détectées par l'utilisateur de cet élément (application);
- des erreurs propres à l'application et aux ressources.

Le présent appendice décrit comment les erreurs génériques définies dans l'élément CMISE peuvent s'appliquer au modèle de gestion du trafic et aux services définis dans la présente Recommandation. Cette description est réalisée avec un exemple; ce scénario n'est pas représentatif de tous les types d'erreurs possibles.

IX.2 Erreurs générées par la machine protocolaire

Ces erreurs résultent d'un viol des transitions d'état décrites pour la machine protocolaire ou des règles d'utilisation des services du protocole.

Comme exemple, au niveau de la couche Transport, des dispositions sont prises pour détecter une erreur de séquençement dans les unités PDU de transport et les procédures de rétablissement après erreur. Au niveau de la couche Présentation, si les données codées reçues n'appartiennent pas à l'ensemble approuvé de syntaxes abstraites (négocié dans le cadre de l'ensemble de contexte de définitions), un abandon sera généré. De plus, si aucune règle d'extensibilité n'est fournie, des abandons de la couche Présentation peuvent se produire si un système envoie par exemple une valeur de type énuméré (cette procédure figure dans un document révisé) à un autre système qui a été amélioré afin de reconnaître cette nouvelle valeur.

La prescription CMIP (protocole commun d'information de gestion) qui dit qu'une association doit être établie avant l'envoi d'unité de données protocolaire PDU CMIP est un exemple de viol d'une règle, qui peut causer une erreur dans l'application. Si cette dernière émet des unités PDU CMIP sans association, il y aura une erreur.

Le paragraphe IX.3 discute l'utilisation des erreurs définies dans l'élément CMISE à employer par l'application.

IX.3 Utilisation d'erreur CMISE

Afin d'illustrer l'utilisation d'erreurs génériques définies dans l'élément CMISE, on considère l'exemple suivant. Le modèle de gestion du trafic définit la classe *tmCircuitEndPointSubgroup* comme une sous-classe de la classe *circuitEndPointSubgroup* définie dans la Recommandation M.3100. Cette sous-classe comporte un attribut appelé *tmSurveillance*. Son comportement est soumis à une contrainte d'intégrité qui satisfait aux conditions de création et de suppression suivantes:

Si la valeur de l'attribut *tmSurveillance* est positionnée sur vrai, alors il doit exister un objet *tmCircuitEndPointSubgroupCurrentData*. La condition de création de cet objet est donc de positionner la valeur de l'attribut *tmSurveillance* sur vrai. Si cet objet est supprimé, alors la valeur de l'attribut doit être positionnée sur faux. En d'autres termes, la condition de suppression de l'objet *currentData* est de positionner la valeur de l'attribut sur faux. S'il existe une erreur dans ce comportement, c'est-à-dire si la valeur de l'attribut est vrai sans que l'objet n'existe, alors la contrainte d'intégrité est violée et la mise en œuvre ne sera pas conforme. Cette remarque est aussi valable pour la situation inverse.

Supposons que le gestionnaire soit intéressé à la collecte des données de surveillance des sous-faisceaux d'extrémité de circuits. Une demande CMIP de positionnement de la valeur d'un attribut peut être émise par l'une des méthodes suivantes:

- 1) demande de modification d'une ou des valeurs d'un ou plusieurs des attributs d'un seul objet géré en utilisant une opération de positionnement non confirmée;
- 2) demande de modification d'une ou des valeurs d'un ou plusieurs des attributs d'un seul objet géré en utilisant une opération de positionnement confirmée;
- 3) demande de modification d'une ou des valeurs d'un ou plusieurs des attributs de plusieurs objets avec délimitation de domaine (indépendamment du filtrage) et en utilisant une opération de positionnement non confirmée;
- 4) demande de modification d'une ou des valeurs d'un ou plusieurs des attributs de plusieurs objets avec délimitation de domaine (indépendamment du filtrage) et en utilisant une opération de positionnement confirmée.

Les deux derniers cas peuvent être séparés plus loin en deux modes: synchronisation atomique et synchronisation la meilleure possible.

Les erreurs susceptibles d'apparaître en positionnant l'attribut *tmSurveillance* sont décrites ci-dessous.

Dans le premier cas, si le système est incapable de créer l'objet de la classe *currentData*, aucune réponse d'erreur ne sera émise. Toutefois, si ce mode de positionnement est choisi, alors le discriminateur de retransmission d'événement doit être configuré de façon à émettre des notifications de création. L'absence de ces notifications indiquera que l'objet *currentData* n'a pas été créé. Il est alors possible d'émettre une commande "lecture de l'attribut" afin de déterminer ce résultat. La représentation de cette erreur sur l'écran de l'exploitant dépasse le cadre de l'interfaçage décrit dans la présente Recommandation. Une autre option est que l'objet *tmCircuitEndPointSubgroup* envoie une notification de changement de valeur d'attribut précisant l'ancienne et la nouvelle valeur de l'attribut *tmSurveillance*.

Si cet attribut est déjà positionné sur vrai et si l'instance de l'objet *currentData* est présente, une demande de positionnement sur la valeur vrai n'aura aucun effet. Le comportement prévu lorsque cet objet existe déjà doit être spécifié.

Dans le deuxième cas, il est possible qu'une réponse d'erreur à la demande de positionnement soit émise. Elle peut correspondre à l'un des codes appropriés suivants: accès refusé, *classInstanceConflict*, *noSuchObjectClass*, *noSuchObjectInstance*, *processingFailure* sans aucune information supplémentaire, ce qui suppose qu'il n'existe aucun modèle de paramètre défini pour cette classe d'objet, *setListError* avec ou sans valeur fournie dans la demande. Si l'opération n'est pas exécutée suite à une limitation des ressources, alors une unité PDU RO-Reject avec le code d'erreur *resourceLimitation* peut être émise. La réception de cette erreur ou la non-réception de la notification de création de l'objet de la classe *currentData* signalera qu'un message d'erreur approprié est envoyé à l'exploitant. Le passage du code d'erreur de l'unité PDU vers l'interface conviviale dépasse le cadre du service RGT.

Le troisième cas est similaire au premier, si ce n'est que l'attribut *tmSurveillance* de plusieurs objets doit être positionné sur vrai, si les objets *currentData* correspondants sont créés. Comme cette demande est non confirmée, des séries de notifications de créations d'objets seront reçues pour les objets *currentData* créés. Ces notifications comporteront les noms des objets *currentData* (et le nom du sous-faisceau *circuitEndPointSubgroup* qui les contient). Elles peuvent alors servir à déterminer si tous les objets *currentData* ont été créés. Ceci suppose que le gestionnaire connaît le nom des sous-faisceaux d'extrémité de circuits pour lesquels des mesures sont requises. Compte tenu de la notification de création, le gestionnaire peut alors fournir à l'exploitant un message convivial (hors du cadre de la présente Recommandation) indiquant si des erreurs ont été commises. L'exploitant peut alors demander un retour en arrière. Cette reprise implique que le passage de l'interface de l'utilisateur à l'interface du protocole CMIP supprime toutes les instances de la classe *currentData* pour lesquelles des notifications de création ont été reçues. Quand la demande de suppression est émise, les contraintes d'intégrité demandent le positionnement de l'attribut *tmSurveillance* de l'objet contenu sur la valeur faux (cela doit être établi dans la corrélation de nom de chaque classe d'objets *currentData*).

Le quatrième cas est similaire au deuxième. Les réponses liées au code *setListError* ou aux notifications de création d'objets peuvent être analysées pour déterminer les sous-faisceaux d'extrémité de circuits pour lesquels les objets *currentData* n'ont pas été créés.

Les descriptions des cas 3 et 4 s'appliquent si l'opération a spécifié que la valeur du paramètre de synchronisation est synchronisation la meilleure possible ou si aucune valeur de ce paramètre n'a été incluse dans la demande. Si on souhaite que les objets *currentData* ne soient créés que quand ils sont tous "créables" alors la synchronisation atomique doit être spécifiée. La méthode mise en œuvre, en termes de verrouillage de l'information jusqu'à ce que toutes les opérations soient terminées, dépasse le cadre du modèle. De plus, si une panne apparaît dans la liaison de communication durant l'un des cas, alors les procédures de rétablissement seront requises pour assurer l'homogénéité de la base MIB. Cette opération dépasse encore le cadre du modèle.

Un autre exemple, dans lequel des erreurs peuvent apparaître, porte sur la gestion de fonctions absentes d'un type particulier d'élément de réseau. L'attribut *destinationAspect* comporte par exemple des valeurs de type énuméré correspondant aux fonctions de type difficile d'accès. Dans toutes les définitions d'attribut, les valeurs peuvent être fournies pour englober un ensemble de types d'élément de réseau avec différentes capacités. Si une demande de positionnement est émise pour positionner la valeur correspondante sur difficile d'accès et si l'élément de réseau ne prend pas cette fonctionnalité en charge, alors la valeur d'erreur CMISE *invalidAttribute* sera émise (en supposant que cette demande était confirmée). Les restrictions appliquées sur les valeurs prises en charge dans une réalisation de produit doivent être spécifiées dans le formulaire MOCS (formulaire de déclaration de conformité d'objets gérés). Cette mention servira à comprendre pourquoi la valeur *invalidAttribute* est reçue même si la valeur utilisée correspond à une valeur valide définie dans la norme.

IX.4 Utilisation des erreurs spécifiques

En plus des erreurs génériques définies par l'élément CMISE, il est possible d'inclure des erreurs propres à la ressource dans le cadre de la définition CMIP. Ces erreurs devront être spécifiées avec les modèles de paramètres et identifiées dans le modèle d'information (elles seront par exemple liées à un attribut pour indiquer qu'il est possible d'émettre un message d'erreur de ce type avec une syntaxe précise lors de la modification de cet attribut).

L'exemple suivant a été imaginé pour illustrer ce point.

Supposons qu'un élément de réseau est désigné pour prendre en charge la commande *cancelTo* et que, suite aux limitations de la ressource, le fournisseur spécifie le nombre maximal d'objets *cancelTo*, susceptibles d'être pris en charge par l'élément de réseau. La demande de création échouera si, suite à l'opération de création, ce nombre est dépassé. Il est possible de prévoir à cet effet une instruction de rejet générique appelée *resourceLimitation*. Par ailleurs, un paramètre d'erreur spécifique appelé *createError* est défini dans la Recommandation M.3100. Si ce paramètre est inclus dans la corrélation de nom de la classe d'objet *cancelTo*, alors une erreur due à une panne de traitement et un entier représentant le nombre maximal d'objet *cancelTo*, susceptibles d'être pris en charge par l'élément de réseau, peuvent être fournis. L'interface de l'utilisateur peut alors transformer cette information en une chaîne significative et l'afficher sur l'écran de l'exploitant.

IX.5 Conclusion

Comme montré précédemment, la plupart des conditions d'erreurs peuvent être représentées par des erreurs génériques définies dans l'élément CMISE en spécifiant les contraintes d'intégrité. De plus, des erreurs spécifiques peuvent être incluses, mais elles doivent l'être dans les modèles de classes d'objets gérés, d'attribut, d'action ou de notification. Des erreurs comme fin de temporisation ou panne de liaison dépassent le cadre du modèle d'information. Elles doivent être déterminées par les spécifications de mise en œuvre de l'interface. Aucune erreur spécifique n'a été identifiée dans le modèle.

APPENDICE X

Hiérarchie des commandes

Comme la Recommandation E.412 ne détaille pas la hiérarchie des commandes, lorsque plusieurs d'entre elles s'appliquent simultanément à un même commutateur, on utilise les règles de détermination de hiérarchie suivantes:

- Puisque la destination est déterminée avant la recherche de sous-faisceaux de circuits, les commandes de destination ont priorité sur toutes les commandes de sous-faisceaux.
- Les commandes de sous-faisceaux s'appliquent alors à deux niveaux: niveau 1: avant la recherche de circuit au repos du sous-faisceau; et niveau 2: après le débordement de tous les circuits du sous-faisceau mais avant la recherche du prochain sous-faisceau de débordement de la chaîne. Les commandes du niveau 1 ont priorité sur celles de niveau 2.

La hiérarchie descendante des commandes du trafic à considérer durant le traitement de l'appel est la suivante:

- | | | |
|--|---|------------------------|
| <ol style="list-style-type: none"> 1) Commande d'indicatif de destination 2) Commande automatique de destination 3) Débordement temporaire vers 4) Annulation du débordement vers 5) Evitement 6) Réservation sélective de circuit 7) Réduction automatique de l'encombrement | <div style="border-left: 1px solid black; border-right: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 20px; height: 100px; margin: 0 auto;"></div> | \Rightarrow niveau 1 |
|--|---|------------------------|

- 8) Annulation du trafic de débordement réacheminé
 - 9) Débordement temporaire depuis
 - 10) Annulation du débordement depuis
- ⇒ niveau 2

Si la priorité entre les commandes n'est pas spécifiée dans le présent appendice, on considérera qu'elle relève de décisions locales.

SÉRIES DES RECOMMANDATIONS UIT-T

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