



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.606.1

(02/2003)

SÉRIE X: RÉSEAUX DE DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS
Réseautage OSI et aspects systèmes – Réseautage

**Technologies de l'information – Protocole de
transport de communications amélioré:
spécification de la gestion de la qualité de
service pour le transport simplex en
multidiffusion**

Recommandation UIT-T X.606.1

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

RÉSEAUX PUBLICS DE DONNÉES	
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés des couches	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

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

**Technologies de l'information – Protocole de transport de communications
amélioré: spécification de la gestion de la qualité de service pour
le transport simplex en multidiffusion**

Résumé

La présente Recommandation | Norme internationale (deuxième partie du protocole de transport de communications amélioré (ECTP)) spécifie un protocole de transport de bout en bout à multidiffusion prenant en charge la négociation, les fonctions de surveillance et de maintien de qualité de service dans une connexion multidébit simplex (1 à n). Ce protocole peut être utilisé pour des applications multimédias tels des services multimédias en temps réel pour la prise en charge des besoins en matière de qualité de service.

Si la gestion de la qualité de service est activée, la négociation des paramètres de qualité de service peut être facultativement effectuée pendant la phase de création de la connexion. L'expéditeur gère alors les valeurs des paramètres à partir de celles proposées par les destinataires. Si la négociation n'est pas activée, l'expéditeur impose des valeurs prédéfinies.

Le niveau de qualité de service obtenu est observé pendant la phase de transfert de données. Chaque destinataire mesure les valeurs de qualité de service effectives et fait rapport de l'état de chaque paramètre à son unité de contrôle "parent" en utilisant des paquets ACK modifiés. L'émetteur compile les valeurs d'état de paramètre indiquées par les destinataires pour obtenir un aperçu de la QS de la connexion. L'expéditeur prend les mesures de maintien de QS, tels l'ajustement du débit de transmission de données, fondées sur l'analyse de la QS de la connexion.

Source

La Recommandation X.606.1 de l'UIT-T, élaborée par la Commission d'études 17 (2001-2004) de l'UIT-T, a été approuvée le 13 février 2003. Un texte identique est publié comme Norme Internationale ISO/CEI 14476-2.

Mots clés

Protocole de transport de communications amélioré (ECTP), protocole de transport de multidiffusion, simplex en multidiffusion, qualité de service (QS).

AVANT-PROPOS

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

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

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

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© UIT 2003

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

TABLE DES MATIÈRES

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives..... 2
3	Définitions..... 2
3.1	Termes définis dans la Rec. UIT-T X.605 ISO/CEI 13252..... 2
3.2	Termes définis dans la Rec. UIT-T X.606 ISO/CEI 14476-1..... 2
3.3	Termes définis dans la présente Recommandation Norme internationale..... 2
4	Abréviations..... 3
4.1	Types de paquet..... 3
4.2	Divers..... 3
5	Conventions..... 3
6	Aperçu général..... 3
7	Composantes de la gestion de QS..... 6
7.1	Élément d'information de connexion..... 6
7.2	Paramètres de QS..... 7
7.3	Élément d'extension de QS..... 8
7.4	Élément accusé de réception..... 9
7.5	Paquets utilisés pour la gestion de la QS..... 10
8	Procédures de gestion de la QS..... 10
8.1	Négociation de la QS..... 11
8.1.1	Procédures de négociation..... 11
8.1.2	Négociation de la QS dans une hiérarchie arborescente..... 12
8.1.3	Négociation MSS..... 12
8.1.4	Réservation des ressources..... 13
8.2	Surveillance de la QS..... 13
8.2.1	Génération de l'ACK..... 14
8.2.2	Mesure des valeurs des paramètres de QS..... 14
8.2.3	Mappage en une valeur d'état d'un paramètre..... 15
8.2.4	Rapport à l'expéditeur..... 15
8.3	Maintien de la QS..... 16
8.3.1	Ajustement du débit de transmission de données..... 17
8.3.2	Pause et reprise de la connexion..... 17
8.3.3	Élimination de l'élément perturbateur..... 18
8.3.4	Terminaison d'une connexion..... 18
9	Temporisations et variables..... 18
9.1	Temporisations..... 18
9.2	Variables opérationnelles..... 19
Annexe A	– Interfonctionnement entre les protocoles ECTP et RSVP pour la réservation des ressources..... 20
A.1	Paramètres de QS ECTP..... 20
A.2	Aperçu général du protocole RSVP..... 20
A.2.1	Objet RSVP SENDER_TSPEC..... 20
A.2.2	Objet RSVP ADSPEC..... 21
A.2.3	Objet RSVP FLOWSPEC..... 21
A.2.4	Interface API RSVP..... 21
A.3	Exemple de mappage des paramètres entre protocoles RSVP et ECTP..... 22
A.4	Scénario d'interfonctionnement entre protocoles ECTP et RSVP..... 22
Annexe B	– Interfaces de programmation d'application..... 25
B.1	Aperçu général..... 25
B.1.1	Fonctions de l'interface API..... 25
B.1.2	Utilisation des fonctions API ECTP..... 25
B.2	Fonctions API ECTP..... 26
B.2.1	Fonction msocket()..... 26
B.2.2	Fonction mbind()..... 27
B.2.3	Fonction maccept()..... 29

	<i>Page</i>
B.2.4 Fonction mconnect()	29
B.2.5 Fonction msend()	30
B.2.6 Fonction mrecv().....	31
B.2.7 Fonction mclose()	31
B.2.8 Fonctions mgetsockopt() et msetsockopt().....	32
B.3 Exemple d'un fichier d'en-tête msocket.h.....	34
Bibliographie.....	37

Introduction

La présente Recommandation | Norme internationale définit le protocole de transport de communications amélioré (ECTP, *enhanced communications transport protocol*), qui est un protocole de transport visant à prendre en charge les applications multidiffusion Internet fonctionnant sur les réseaux pouvant assurer la multidiffusion. Le protocole ECTP fonctionne sur les réseaux IPv4/IPv6 ayant une capacité de transmission multidiffusion IP au moyen de protocoles de routage multidiffusion IP et IGMP, comme indiqué à la Figure 1. Le protocole ECTP peut être configuré en mode UDP.

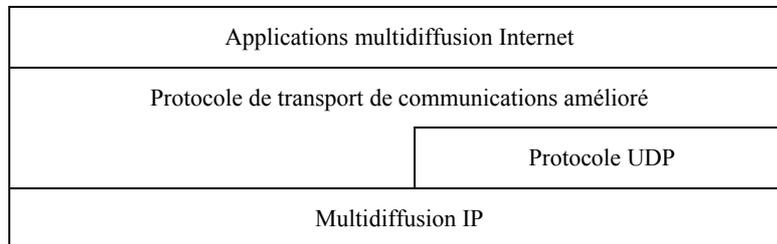


Figure 1 – Modèle ECTP

Le protocole ECTP est destiné à prendre en charge des connexions de multidiffusion étroitement gérées dans les applications simplex, duplex et N-plex. Cette partie du protocole (partie 2) définit les fonctions de gestion de la QS en vue d'une gestion stable de la qualité de service pour les utilisateurs de la connexion en mode simplex à multidiffusion. Cette fonction de gestion de la QS peut être assurée au moyen d'opérations de négociation, de surveillance et de maintien de la qualité de service. Les procédures protocolaires de la gestion de la fiabilité pour le transport simplex en multidiffusion sont définies dans la partie 1 de l'ECTP (Rec. UIT-T X.606 | ISO/CEI 14476-1) qui fait partie intégrante de la présente Recommandation | Norme internationale. D'autres parties de la norme définiront les procédures de commande et les fonctions associées de gestion de la QS pour le mode duplex (X.ectp-3 | ISO/CEI 14476-3 et X.ectp-4 | ISO/CEI 14476-4) et le mode N-plex (X.ectp-5 | ISO/CEI 14476-5 et X.ectp-6 | ISO/CEI 14476-6).

En mode ECTP, tous les membres potentiels sont enrôlés dans un groupe de multidiffusion avant la création d'une connexion ou d'une session. Ces membres définissent un groupe enrôlé. Chaque destinataire du groupe enrôlé est dénommé destinataire enrôlé. Pendant le processus d'enrôlement, chaque membre sera authentifié. Les informations de groupe, y compris la clé de groupe, les adresses multidiffusion IP et les numéros de ports, seront distribuées aux membres enrôlés pendant le processus d'enrôlement. Une connexion ECTP est créée pour ces membres du groupe enrôlé.

L'expéditeur est au centre des communications du groupe multidiffusé. Dans la connexion multidiffusion simplex, le rôle de propriétaire de la connexion est attribué à un seul expéditeur appelé propriétaire principal (TO, *top owner*) dans la présente Spécification. Le propriétaire de la connexion est responsable de la gestion globale de la connexion en ce sens qu'il gère les opérations de création et de fin de connexion, de pause et de reprise de connexion, de participation à une connexion et de sortie.

L'expéditeur déclenche le processus de création de connexion. Une partie ou la totalité des destinataires enrôlés participeront à la connexion et seront appelés "destinataires actifs". Les destinataires actifs à ce stade peuvent participer à la négociation de la qualité de service souhaitée pour la session. Tout destinataire enrôlé qui n'est pas alors actif peut participer à la connexion en tant que participant tardif, mais il devra accepter la QS établie. Un destinataire actif peut quitter la connexion.

Après création de la connexion, l'expéditeur commence à émettre des données multidiffusées. Pendant que la connexion est active, l'expéditeur observe l'état de la session via des paquets de réaction provenant des destinataires actifs.

L'expéditeur peut prendre un certain nombre de mesures si les problèmes dans le réseau (encombrement sévère par exemple) sont signalés par les destinataires actifs. Parmi ces mesures, citons l'ajustement du débit de transmission de données, la suspension temporaire de la transmission de données multidiffusées ou en dernier ressort, la terminaison de la connexion.

La présente spécification de gestion de la QS peut être utilisée dans les applications de multidiffusion pour lesquelles on souhaite la prise en charge de diverses exigences en matière de QS et les différents modèles de facturation/taxation associés.

**NORME INTERNATIONALE
RECOMMANDATION UIT-T**

**Technologies de l'information – Protocole de transport de communications
amélioré: spécification de la gestion de la qualité de service pour
le transport simplex en multidiffusion**

1 Domaine d'application

La présente Recommandation | Norme internationale fait partie intégrante des Recommandations UIT-T de la série X.606.x | ISO/CEI 14476 "Protocole ECTP: protocole de transport de communications amélioré", qui constitue une famille de spécifications de protocole spécialement conçues pour prendre en charge les services de transport avec multidiffusion.

La Rec. UIT-T X.606 | ISO/CEI 14476-1 contient les spécifications de diverses opérations protocolaires pour le transport simplex en multidiffusion. Ces opérations portent sur la gestion des connexions telle la création/termination de connexion, la pause/reprise de connexion, la gestion des membres tels les participants tardifs, l'abandon par l'utilisateur et le suivi des participants, le contrôle d'erreur dans le transport de données multidiffusées telle la détection/correction d'erreur.

Cette partie de la Recommandation | Norme internationale contient une spécification de la gestion de la QS visant à obtenir la qualité de service souhaitée sur une connexion de transport simplex en multidiffusion.

La présente Spécification décrit les opérations de gestion de QS suivantes:

a) *Négociation de la QS*

En ce qui concerne la négociation de la QS, on suppose dans la présente Spécification que le niveau de QS souhaité pour un service d'application de multidiffusion peut être exprimé par un ensemble de paramètres de QS. La négociation de la QS est réalisée via l'échange de paquets de commande entre expéditeur et destinataires. L'expéditeur propose des valeurs cibles pour les paramètres de QS déterminés à partir des exigences associées à l'application, ensuite chaque destinataire peut proposer des valeurs modifiées fondées sur les capacités de son propre système et/ou du réseau. L'expéditeur procède à un choix en fonction des valeurs modifiées proposées par les destinataires. Les valeurs cibles des paramètres de QS peuvent être utilisées comme paramètres d'entrée pour la réservation des ressources de réseau.

b) *Surveillance de la QS*

La gestion de la QS dans le protocole ECTP se fonde sur la réaction des destinataires par paquets de gestion. Les messages de réaction provenant des destinataires permettent à l'expéditeur de connaître le nombre de destinataires actifs et également de surveiller l'état de la connexion utilisée pour le transport de données multidiffusées. La surveillance de la QS a pour but de permettre à l'expéditeur de faire des diagnostics sur l'état de la connexion en termes de valeurs de paramètres de QS et ainsi de prendre les mesures nécessaires pour maintenir la connexion au niveau de QS souhaité. L'état de la connexion surveillée sera signalé à l'application côté expéditeur. L'information acheminée doit permettre l'établissement de statistiques utiles pour la facturation par exemple.

c) *Maintien de la QS*

Sur la base des informations communiquées par réaction par les destinataires, l'expéditeur prend les mesures nécessaires pour maintenir la QS de la connexion au niveau souhaité. Parmi ces mesures figurent l'ajustement du débit de transmission de données, la pause et la reprise de la connexion, l'éjection de l'élément perturbateur et les opérations de terminaison de la connexion. Ces fonctions de surveillance et de maintien de la QS, basées sur l'état des paramètres observés, permettent une gestion des encombrements par le débit.

La présente Recommandation | Norme internationale fait partie intégrante des Recommandations UIT-T de la série X.606.x | ISO/CEI 14476, qui comporte 6 parties. Toutes les composantes du protocole, y compris les formats des paquets et les procédures protocolaires spécifiés dans la Rec. UIT-T X.606 | ISO/CEI 14476-1, sont également valables dans la présente Recommandation | Norme internationale.

2 Références normatives

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

- Recommandation UIT-T X.601 (2000), *Cadre général des communications entre homologues multiples*.
- Recommandation UIT-T X.605 (1998) | ISO/CEI 13252:1999, *Technologies de l'information – Définition du service de transport de communications amélioré*.
- Recommandation UIT-T X.606 (2001) | ISO/CEI 14476-1:2002, *Technologies de l'information – Protocole de transport de communications amélioré: spécification du transport simplex en multidiffusion*.

3 Définitions

3.1 Termes définis dans la Rec. UIT-T X.605 | ISO/CEI 13252

La présente Recommandation | Norme internationale est fondée sur les concepts développés pour le service de transport de communications amélioré (Rec. UIT-T X.605 | ISO/CEI 13252).

- a) paramètres de QS;
- b) négociation de QS;
- c) arbitrage de QS.

3.2 Termes définis dans la Rec. UIT-T X.606 | ISO/CEI 14476-1

La présente Recommandation | Norme internationale s'appuie sur les concepts et les termes définis dans la spécification du transport simplex en multidiffusion utilisant le protocole ECTP (Rec. UIT-T X.606 | ISO/CEI 14476-1):

- a) application;
- b) paquet;
- c) expéditeur;
- d) destinataire;
- e) arborescence;
- f) parent;
- g) enfant.

3.3 Termes définis dans la présente Recommandation | Norme internationale

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent:

- a) **surveillance de la QS**: opération protocolaire qui est utilisée pour établir un diagnostic de l'état actuel de la connexion. Pour la surveillance de la QS, chaque destinataire doit mesurer les valeurs observées des paramètres et les communiquer à l'expéditeur. L'expéditeur compile des informations d'état communiquées par les destinataires.
- b) **maintien de la QS**: opération protocolaire qui est utilisée pour maintenir l'état de la connexion avec le niveau de QS souhaité. L'expéditeur prend les mesures de maintien de la QS sur la base des informations d'état résultant des observations.

4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale les abréviations suivantes s'appliquent.

4.1 Types de paquet

ACK	Accusé de réception (<i>acknowledgment</i>)
CC	Confirmation de création de connexion
CR	Demande de création de connexion (<i>connection creation request</i>)
CT	Fin de connexion (<i>connection termination</i>)
DT	Données (<i>data</i>)
HB	Pulsation (<i>heartbeat</i>)
JC	Confirmation de participation tardive (<i>late join confirm</i>)
JR	Demande de participation tardive (<i>late join request</i>)
LR	Demande de sortie (<i>leave request</i>)
ND	Données nulles (<i>null data</i>)
RD	Données de retransmission (<i>retransmission data</i>)

4.2 Divers

API	Interface de programmation d'application (<i>application programming interface</i>)
CHQ	Qualité la plus élevée gérée (<i>controlled highest quality</i>)
Diffserv	Services différenciés (<i>differentiated services</i>)
ECTP	Protocole de transport de communications amélioré (<i>enhanced communications transport protocol</i>)
ECTS	Services de transport de communications améliorés (<i>enhanced communications transport services</i>)
IP	Protocole Internet (<i>Internet protocol</i>)
LQA	Qualité la plus faible autorisée (<i>lowest quality allowed</i>)
MSS	Taille maximale de segment (<i>maximum segment size</i>)
OT	Cible opérante (<i>operating target</i>)
QS	Qualité de service
RSVP	Protocole de réservation de ressources (<i>resource reservation protocol</i>)

5 Conventions

Dans la présente Recommandation | Norme internationale, les mots clés "DOIT", "REQUIS", "DEVRA", "NE DOIT PAS", "DEVRAIT", "NE DEVRAIT PAS", "PEUT" et "FACULTATIF" indiquent le degré de contrainte d'une prescription pour les implémentations de protocole ECTP conformes.

6 Aperçu général

La présente Recommandation | Norme internationale spécifie la gestion de QS pour des connexions de transport (simplex) en multidiffusion un-plusieurs. La présente Spécification décrit les opérations de gestion de QS suivantes:

- 1) négociation de QS, y compris la réservation des ressources de réseau;
- 2) surveillance de la QS;
- 3) maintien de la QS.

Dans la phase de création d'une connexion, l'expéditeur indique au destinataire si la gestion de QS est activée. Si tel est le cas, l'expéditeur doit également spécifier s'il y aura négociation de QS pour la connexion considérée. Les opérations de surveillance et de maintien de la QS sont exécutées seulement si la gestion de QS est activée.

La Figure 2 illustre ces opérations de gestion de QS pour une connexion simplex en multidiffusion. Dans cette figure, les opérations protocolaires marquées par des pointillés sont spécifiées dans la Rec. UIT-T X.600 | ISO/CEI 14476-1.

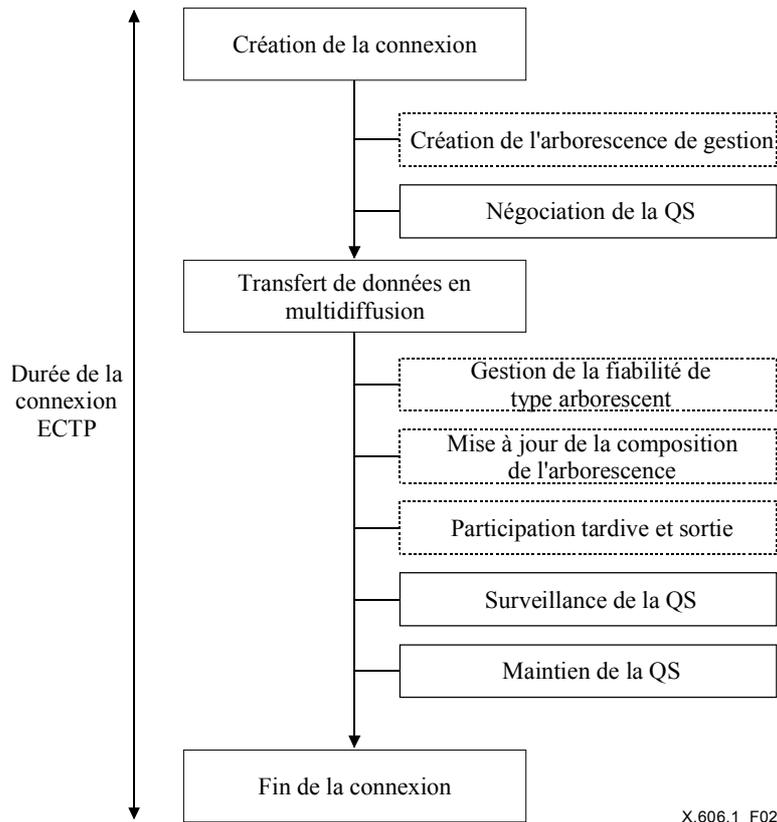


Figure 2 – Gestion de la QS dans le cadre du protocole ECTP

En général, la QS représente la qualité de service nécessaire à une réception satisfaisante des données d'application par un destinataire; par exemple, celle qui permet une présentation audio/vidéo avec la qualité souhaitée. Dans la présente Spécification, on suppose que les exigences de QS associées à une application sont exprimées en termes de paramètres de QS tels le débit, le temps de transit, la gigue de temps de transit et le taux de perte de données. En fonction des exigences propres à une application, certains de ces paramètres de QS peuvent ne pas être utilisés dans la connexion. Ainsi, un service en temps non réel peut ne pas imposer des exigences en matière de temps de transit.

A partir des exigences associées aux applications, l'expéditeur déterminera les valeurs cibles de chaque paramètre de QS. La façon de faire correspondre ces exigences avec les valeurs cibles des paramètres est hors du domaine d'application de la présente Spécification. Des programmes d'application pourraient être utilisés pour établir cette correspondance.

La négociation de QS est effectuée pendant la phase de création de la connexion. L'expéditeur propose par multidiffusion des valeurs cibles désirées pour chaque paramètre de QS à tous les destinataires. Pour ce qui est du débit, trois valeurs cibles sont spécifiées: CHQ (*controlled highest quality*: qualité maximale contrôlée), OT (*operating target*: cible opérationnelle) et LQA (*lowest quality allowed*: qualité minimale autorisée). Pour les autres paramètres tels le temps de transit, la gigue de temps de transit et le taux de perte de données, seules deux valeurs cibles sont spécifiées, à savoir: OT et LQA.

Lorsque la négociation de QS est activée, chaque destinataire peut proposer des modifications aux valeurs des paramètres proposées par l'expéditeur. Ces valeurs modifiées seront choisies en prenant en considération la capacité du système côté expéditeur et les environnements réseau. Les restrictions suivantes sont imposées à la modification des valeurs des paramètres par les destinataires:

- 1) les valeurs d'OT ne doivent pas être modifiées par les destinataires;
- 2) les valeurs modifiées par les destinataires doivent être comprises entre les valeurs LQA et les valeurs CHQ proposées par l'expéditeur.

Les valeurs des paramètres modifiées par les destinataires sont communiquées à l'expéditeur via des messages ACK. L'expéditeur choisit pour chaque destinataire des valeurs des paramètres dans une fourchette de valeurs établie en commun.

La Figure 3 illustre schématiquement un exemple de négociation de QS dans le cas du protocole ECTP. A partir des exigences associées à l'application, un ensemble de valeurs cibles de paramètre de QS sera établi par l'expéditeur. L'expéditeur fait connaître aux destinataires des valeurs cibles (étape 1). Sur la base de ces valeurs cibles, chaque destinataire commence alors à réserver les ressources nécessaires à l'aide du protocole RSVP ou du service Diffserv (étape 2). Si la négociation de QS est activée pour la connexion, chaque destinataire peut proposer des valeurs modifiées des paramètres de QS (étape 3). Sur la base de ces valeurs modifiées, l'expéditeur choisit alors des valeurs pour les paramètres (étape 4). Les valeurs ainsi choisies sont communiquées au destinataire via des paquets HB ou JC subséquents, et seront utilisés pour la surveillance et le maintien de la QS.

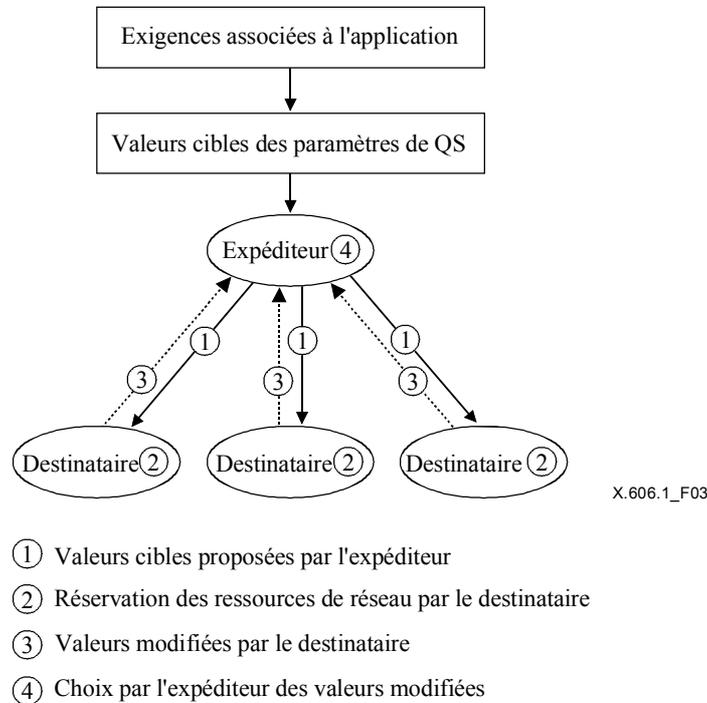


Figure 3 – Négociation de QS

Après création de la connexion ECTP et si la gestion de QS est activée, les opérations de surveillance et de maintien de la QS sont exécutées pour la transmission de données multidiffusées. En ce qui concerne la surveillance de la QS, chaque destinataire doit mesurer les valeurs observées des paramètres. Sur la base des valeurs mesurées et des valeurs négociées, un destinataire détermine une valeur d'état de paramètre pour chaque paramètre sous forme d'un entier: normal (0), raisonnable (1), éventuellement anormal (2), ou anormal (3). Ces valeurs d'état sont communiquées à l'expéditeur via des paquets ACK.

L'expéditeur compile les valeurs d'état des paramètres communiquées par les destinataires. Si une arborescence de contrôle est utilisée, chaque nœud LO parent compile les valeurs mesurées signalées par ses enfants, et retransmet cette ou ces valeurs compilées à son propre parent au moyen de paquets ACK.

La Figure 4 illustre les opérations de surveillance et de maintien de QS décrites dans la présente Spécification.

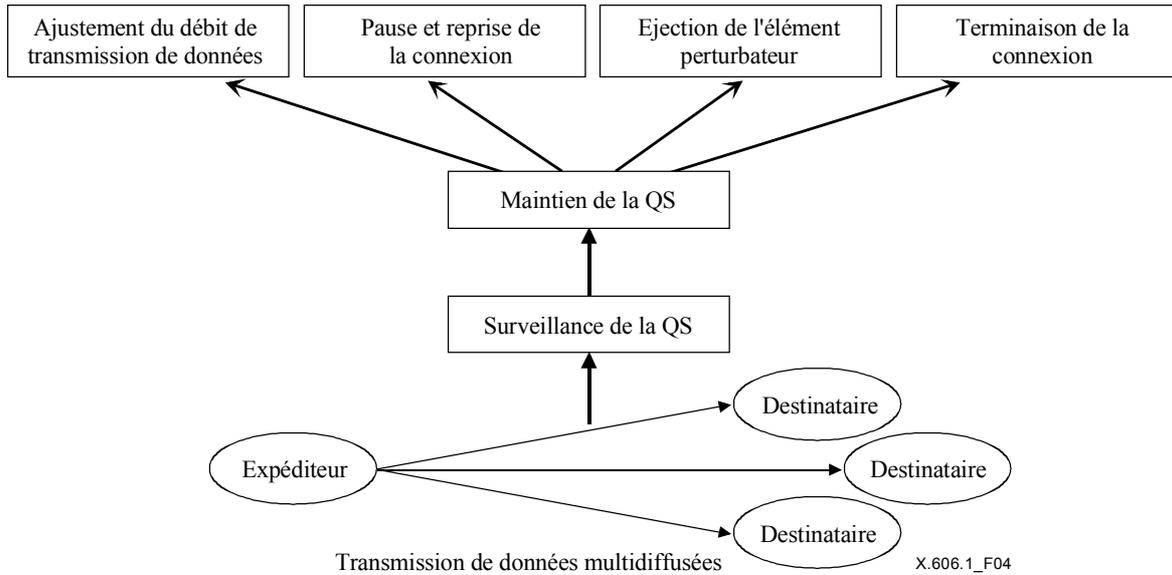


Figure 4 – Schéma conceptuel des opérations de surveillance et de maintien de QS

L'expéditeur prend les mesures nécessaires de maintien de la QS de la connexion au niveau désiré, en tenant compte des valeurs d'état observées. Des règles spécifiques préconfigurées permettant de déclencher les actions de maintien de la QS tels l'ajustement de débit de transmission de données, la pause et la reprise de la connexion, l'élimination de l'élément perturbateur et la terminaison de la connexion. Ces actions seront prises après observation du nombre de destinataires qui se trouvent dans un "état anormal" ou éventuellement "anormal".

7 Composantes de la gestion de QS

Le présent article décrit les composantes du protocole ECTP nécessaires aux opérations de gestion de la QS. Toutes les composantes sont issues de celles déjà définies dans la Rec. UIT-T X.606 | ISO/CEI 14476-1.

7.1 Élément d'information de connexion

La Figure 5 représente l'élément d'information de connexion spécifié dans la Rec. UIT-T X.606 | ISO/CEI 14476-1.

0	8	16	24	31	
Elément suivant	Version	Indicateurs	Option de config. d'arb.	Nbre max. de niveaux d'arb.	Nombre maximal d'enfants
Délai de création de la connexion			Taille de la suite binaire ACK	Réservé	

Figure 5 – Élément d'information de connexion

Pour la gestion de la QS, l'expéditeur ECTP spécifie les trois champs suivants dans l'octet "indicateur":

	7	6	5	4	3	2	1	0
	Reserved	Retx	N	QS	CT			

- 1) *QS* – il s'agit d'un bit fanion qui indique si la gestion de QS est activée (1) ou non (0) pour la connexion considérée. Si ce bit est mis à "1", toutes les procédures de gestion de QS sont activées. La valeur par défaut est "0";

- 2) *N (négociation)* – il s'agit d'un bit fanion qui indique si la négociation de QS est activée (1) ou non (0) dans la connexion. Si le bit est mis à "1", chaque destinataire est autorisé à proposer ses propres valeurs de paramètre. La valeur par défaut est "0";
- 3) *Retx (retransmission)* – est un bit fanion qui indique si des retransmissions par le parent sont exécutées (0) ou non (1). Si ce bit est mis à "1", l'expéditeur ou les parents ne doivent pas retransmettre les paquets RD, même si des demandes de retransmission par paquets ACK sont reçues. La valeur par défaut est "0".

Le bit QS doit être mis à "1" (QS activée) avant que le bit N soit valide. Il existe trois cas possibles:

- a) le bit QS est mis à "1" et le bit N est à "0", ce qui indique que la QS est utilisée dans la connexion et que les valeurs de QS seront imposées par l'expéditeur. Les destinataires ne peuvent les négocier;
- b) les deux bits sont mis à "1", ce qui indique que la QS est utilisée dans la connexion et que les valeurs des paramètres de QS peuvent être négociées entre les destinataires et l'expéditeur;
- c) le bit de QS est mis à "0", ce qui indique que la QS n'est pas utilisée dans la connexion. Dans ce cas, le bit N n'est pas utilisé.

Le positionnement du bit Retx (bit de retransmission) peut être fait de manière indépendante du positionnement du bit QS. Selon toute probabilité, pour les applications à flux médias continu en temps réel, il ne sera pas nécessaire d'avoir des reprises après erreur fondées sur la retransmission, mais ces applications nécessiteront l'utilisation de fonctions de gestion de QS. Même dans ce cas, les paquets ACK sont toujours utilisés pour acheminer des informations d'état de la connexion.

7.2 Paramètres de QS

Dans la présente Spécification, les quatre paramètres suivants de QS sont définis:

- 1) débit (octets/s);
- 2) temps de transit (ms);
- 3) gigue de temps de transit (ms);
- 4) taux de perte de données (%).

Le débit représente le volume de données d'application sortant pendant un certain intervalle de temps. Par débit cible on entend la valeur de débit requise pour la présentation voulue des données d'application. Les applications génèrent des données multidiffusées et l'expéditeur ECTP les transmettent, sur la base de la ou des valeurs cibles de débit. Le débit réel en réception des données côté destinataire dépendra du débit de transmission de données, des conditions de réseau et de la capacité du système d'extrémité, etc.

En ce qui concerne le débit, l'expéditeur doit configurer les valeurs cibles suivantes:

- 1) débit CHQ;
- 2) débit OT;
- 3) débit LQA.

Les inégalités suivantes entre ces valeurs doivent être vérifiées: débit LQA \leq débit OT \leq débit CHQ.

Le temps de transit représente le temps de transmission de bout en bout entre un expéditeur et un récepteur. Pour obtenir la présentation voulue des données multidiffusées, l'expéditeur peut configurer les valeurs cibles suivantes:

- 1) temps de transit OT;
- 2) temps de transit LQA.

Les inégalités suivantes entre ces valeurs doivent être vérifiées: temps de transit OT \leq temps de transit LQA.

La gigue de temps de transit représente les variations des valeurs du temps de transit. Pour obtenir la présentation voulue des données, l'expéditeur peut configurer les valeurs cibles suivantes:

- 1) gigue de temps de transit OT;
- 2) gigue de temps de transit LQA.

Les inégalités suivantes entre ces valeurs doivent être respectées: gigue de temps de transit OT \leq gigue de temps de transit LQA.

Le taux de perte de données est défini comme étant le rapport du volume de données perdues sur le volume total de données transmises. Pour obtenir la présentation voulue des données, l'expéditeur peut configurer les valeurs cibles suivantes:

- 1) taux de perte OT;
- 2) taux de perte LQA.

Les inégalités suivantes entre ces valeurs doivent être respectées: $\text{taux de perte OT} \leq \text{taux de perte LQA}$.

7.3 Elément d'extension de QS

Pour la gestion de QS, l'élément d'extension de QS est un élément nouveau défini dans la présente Spécification. La liste de tous les éléments d'extension utilisés dans le protocole ECTP est la suivante.

Tableau 1 – Tableau de codage des éléments d'extension ECTP

Elément	Codage
Information de connexion	0001
Accusé de réception	0010
Membre de l'arborescence	0011
Horodate	0100
QS	0101
Pas d'élément	0000

L'élément d'extension de QS spécifie la taille maximale de segment (MSS, *maximum segment size*) et les valeurs cibles des paramètres de QS ECTP décrits au § 7.2. Comme le montre la Figure 6, l'élément QS occupe 28 octets:

0	8	16	24	31
Elément suivant	Version	Indicateurs de QS	Taille maximale du segment	
Débit CHQ				
Débit OT				
Débit LQA				
Temps de transit OT			Temps de transit LQA	
Gigue de temps de transit OT			Gigue de temps de transit LQA	
Taux de perte OT		Taux de perte LQA		Réservé

Figure 6 – Elément d'extension de QS

Les paramètres suivants sont spécifiés:

- a) *élément suivant* – Indique le type de l'élément suivant qui suit immédiatement l'élément de QS considéré;
- b) *version* – Définit la version actuelle du segment considéré, commence à "1";
- c) *indicateurs de QS* – Il s'agit d'un octet fanion qui spécifie si chaque paramètre de QS et la MSS sont utilisés dans la connexion. Le codage de cet octet est indiqué ci-dessous. Si un bit est mis à "1", le paramètre de QS correspondant ou la taille MSS sera utilisé. La valeur par défaut de chaque bit est "0".

7	6	5	4	3	2	1	0
Réservé	E	D	C	B	A		

- 1) A – débit;
- 2) B – temps de transit;
- 3) C – gigue de temps de transit;
- 4) D – taux de perte de données;
- 5) E – taille maximale de segment (MSS);

- 6) *Réservé* – réservé à une utilisation future.
- d) *Taille maximale de segment (MSS)* – Représente la taille maximale d'un segment ou d'un paquet ECTP en octets. Si le bit "E" des fanions de QS sont mis à "1", la taille *MSS* est sujette à négociation (voir § 8.1.3). Dans les autres cas, la valeur *MSS* par défaut utilisée sera de 1024.
- e) *Valeurs de débit* – Chaque valeur est un entier sans signe de 32 bits, l'unité étant l'octet par seconde. Les valeurs cibles suivantes sont valides seulement si le bit "A" des fanions de QS sont mis à "1":
- 1) débit CHQ – limite supérieure de débit;
 - 2) débit OT – débit cible pour l'affichage souhaité des données multidiffusées;
 - 3) débit LQA – limite inférieure de débit.
- f) *Valeurs de temps de transit* – Chaque valeur est un entier sans signe de 16 bits exprimant le temps en millisecondes. Les valeurs suivantes sont valides seulement si le bit "B" des indicateurs de QS est mis à "1":
- 1) temps de transit OT – temps de transit cible pour la présentation voulue des données multidiffusées;
 - 2) temps de transit LQA – temps de transit maximal autorisé.
- g) *Valeurs de gigue de temps de transit* – Chaque valeur est représentée par un entier sans signe de 16 bits exprimant un temps en millisecondes. Les valeurs cibles suivantes sont valides seulement si le bit "C" des indicateurs de QS est mis à "1":
- 1) gigue de temps de transit OT – objectif de gigue de temps de transit pour l'affichage souhaité des données multidiffusées;
 - 2) gigue de temps de transit LQA – gigue de temps de transit maximal autorisé.
- h) *Valeurs de taux de perte de données* – Chaque valeur est représentée par un entier sans signe de 8 bits allant de 0 à 100 en pour cent. Les valeurs cibles suivantes sont valides seulement si le bit "D" des indicateurs de QS est mis à "1":
- 1) taux de perte OT – taux de perte cible pour la présentation voulue des données multidiffusées;
 - 2) taux de perte LQA – taux de perte maximale autorisé.
- i) *Réservé* – réservé à une utilisation future.

L'élément de QS est utilisé par l'expéditeur pour informer les destinataires des valeurs cibles des paramètres de QS au moyen d'un paquet CR dans la phase de création de la connexion. Lors de la négociation de QS, l'élément QS est également utilisé lorsqu'un destinataire propose ses propres valeurs modifiées à l'expéditeur. Les valeurs de QS négociées seront annoncées aux participants tardifs via le paquet JC et aux destinataires existants via les paquets HB.

Ces valeurs de QS sont également utilisées dans les opérations de surveillance et de maintien de la QS.

7.4 Élément accusé de réception

Pour la surveillance de la QS, il est demandé à chaque destinataire de mesurer les valeurs des paramètres qu'il observe. Une valeur de paramètre mesurée est mappée en une valeur d'état de paramètre. Une valeur d'état de paramètre est un entier 0, 1, 2, ou 3. Une valeur élevée d'état indique un état plus médiocre de la connexion.

Les valeurs d'état sont communiquées à l'expéditeur via des paquets ACK. L'élément accusé de réception du paquet ACK contient les valeurs d'état des paramètres de QS utilisées dans la connexion.

L'élément accusé de réception spécifié dans la Rec. UIT-T X.606 | ISO/CEI 14476-1 est représenté ci-dessus. Dans la figure, l'octet "état des paramètres" est nouvellement défini dans la présente Spécification.

	0	8	16	24	31
Élément suivant	Version	Longueur de suite binaire valide	Réservé	Etat paramètre	
Numéro de séquence la plus faible					
Suite binaire					

Figure 7 – Élément accusé de réception

L'octet "état paramètre" a la structure suivante:

7	6	5	4	3	2	1	0
D		C		B		A	

- a) *A* – deux bits indiquant la valeur d'état du débit mesuré;
- b) *B* – deux bits indiquant la valeur d'état pour le temps de transit mesuré;
- c) *C* – deux bits indiquant la valeur d'état pour la gigue de temps de transit mesurée;
- d) *D* – deux bits indiquant la valeur d'état du taux de perte de paquet mesuré.

Les valeurs que peuvent prendre les deux bits d'indication d'état sont les suivantes:

- a) 00 – indique une valeur d'état "0";
- b) 01 – indique une valeur d'état "1";
- c) 10 – indique une valeur d'état "2";
- d) 11 – indique une valeur d'état "3".

Le mappage détaillé d'une valeur de paramètre mesurée en une valeur d'état est décrit au § 8.2.3.

7.5 Paquets utilisés pour la gestion de la QS

Le Tableau 2 contient la liste des paquets ECTP utilisés pour la gestion de la QS.

Tableau 2 – Paquets ECTP utilisés pour la gestion de QS

Type de paquet	Élément extension				
	Information de connexion	Appartenance à l'arborescence	Accusé de réception	Horodate	QS
CR	O				O
CC		O			O
HB		O		O	O
ACK		O	O	O	
JC	O				O

Le paquet CR contient un élément de QS. Il est utilisé par l'expéditeur pour proposer (ou imposer) les valeurs cibles des paramètres de QS utilisés dans la connexion. Ces valeurs peuvent être référencées par des mécanismes de réservation de ressources et des protocoles, tels le protocole RSVP, s'ils sont activés dans le réseau. Si la négociation de QS est activée, chaque destinataire répond à l'expéditeur en proposant ses propres valeurs de paramètres de QS, via un paquet CC. L'expéditeur fait son choix compte tenu des propositions renvoyées, et les valeurs des paramètres de QS qu'il choisit seront indiquées aux destinataires via des paquets HB. Les valeurs cibles des paramètres de QS en cours d'utilisation dans la connexion (imposées ou négociées) sont notifiées aux participants tardifs via des paquets JC (voir § 8.1).

Les opérations de surveillance et de maintien de la QS se font par référence aux valeurs cibles ou négociées. Les paquets ACK sont utilisés pour acheminer les valeurs d'état des paramètres QS constatés côté destinataire (voir § 8.2).

8 Procédures de gestion de la QS

Dans le cas du protocole ECTP, la gestion de la QS fait appel aux opérations suivantes:

- 1) négociation de la QS, éventuellement avec réservation des ressources réseau;
- 2) surveillance de la QS;
- 3) maintien de la QS.

La négociation de QS est effectuée pendant la phase de création de la connexion, tandis que les opérations de surveillance et de maintien de la QS sont effectuées pendant la phase de transmission de données.

Lorsque la gestion de QS d'une connexion est activée, les opérations de surveillance et de maintien de la QS sont effectuées par défaut. Par ailleurs, la négociation de QS n'est activée que si le bit *N* de l'octet "indicateurs" de l'élément d'information de connexion est mis à "1".

8.1 Négociation de la QS

L'expéditeur ECTP transmet un paquet CR à l'intention de tous les destinataires pour déclencher la phase de création de la connexion. Le paquet CR contient les valeurs cibles proposées (ou imposées) de chaque paramètre de QS tels CHQ, OT et LQA. Chaque destinataire peut utiliser ces valeurs cibles pour réserver les ressources nécessaires (voir § 8.1.4). Si la négociation de QS est activée dans la connexion, les procédures de négociation sont elles-mêmes activées (voir § 8.1.1). Les valeurs cibles imposées ou négociées sont alors utilisées pour la surveillance et le maintien de la QS (voir § 8.2 et 8.3).

Si la négociation de QS est activée pour la connexion, chaque destinataire peut proposer une nouvelle valeur modifiée en réaction à une valeur cible de paramètre proposée par l'expéditeur. Pour proposer une nouvelle valeur, le destinataire doit pouvoir identifier les ressources système ou les ressources de réseau à utiliser. Par exemple, une valeur modifiée de débit peut être évaluée à partir des débits de ligne des liaisons de transmission accessibles côté destinataire (par exemple, DSL, câblo-modem, réseaux hertziens, etc.). La valeur modifiée peut également être déterminée en prenant en considération les besoins de l'utilisateur final au niveau du site du destinataire. Il est possible pour un serveur d'extrémité d'utiliser un logiciel pour déterminer la valeur modifiée des paramètres pour négociation, fondée sur les ressources du réseau et les ressources du système disponibles ainsi que sur les besoins de l'utilisateur final. Toutefois, dans des scénarios réels, il n'est pas toujours facile d'identifier précisément la capacité en ressources des réseaux associés à un destinataire. De même, au moins dans un futur proche, la négociation de QS sera effectuée sur la base des besoins des utilisateurs finaux au niveau de l'application et non sur la base de la capacité du système du serveur d'extrémité.

Dans la présente Spécification, l'expéditeur doit indiquer via l'élément d'extension de QS si chaque paramètre de QS peut faire l'objet d'une négociation (voir § 7.3). Pour les paramètres négociables, un destinataire peut proposer des valeurs modifiées. Si un destinataire ne souhaite pas modifier un paramètre de QS, il renverra simplement le même élément de QS qu'il a reçu de l'expéditeur.

8.1.1 Procédures de négociation

Si la négociation de QS est activée dans la connexion, chaque destinataire répond à l'expéditeur par un paquet CC contenant les valeurs cibles modifiées des paramètres de QS respectifs.

Le présent paragraphe décrit les procédures de négociation de QS du paramètre débit, qui a trois valeurs cibles: LQA, OT et CHQ. Les procédures de négociation des autres paramètres tels le temps de transmission, la gigue et le taux de perte sont identiques aux procédures de négociation du paramètre débit à ceci près que ces paramètres n'ont pas de valeur CHQ.

Pendant la négociation de QS, les destinataires ne doivent pas modifier la valeur cible OT de chaque paramètre.

Les procédures détaillées de négociation des paramètres de QS sont décrites ci-dessous et illustrées à la Figure 8.

- 1) L'expéditeur propose des valeurs cibles de paramètre:
à partir des besoins de l'application, l'expéditeur détermine les valeurs cibles des paramètres: LQA₀, OT₀, CHQ₀, où LQA₀ < OT₀ < CHQ₀, et transmet ensuite un paquet CR avec l'élément extension de QS à tous les destinataires.
- 2) Les destinataires modifient les valeurs des paramètres:
en réponse aux valeurs cibles proposées par l'expéditeur, chaque destinataire *i* propose les valeurs modifiées: LQA_{*i*} et CHQ_{*i*}. La valeur cible OT₀ ne doit pas être modifiée. Ainsi, les inégalités suivantes sont imposées: LQA₀ < LQA_{*i*} < OT₀ < CHQ_{*i*} < CHQ₀ pour chaque destinataire *i*. Chaque destinataire remet les valeurs modifiées à l'expéditeur via un paquet CC.
- 3) L'expéditeur procède à un arbitrage entre les valeurs modifiées des paramètres:
l'expéditeur procède à un arbitrage entre les valeurs modifiées des paramètres proposées par les destinataires comme suit:

$$CHQ_{\min} = \min CHQ_i, \text{ pour chaque destinataire } i;$$

$$LQA_{\max} = \max LQA_i, \text{ pour chaque destinataire } i.$$

LQA_{max} et CHQ_{min} sont les valeurs négociées des paramètres qui résultent de la négociation de QS.

- 4) L'expéditeur annonce les valeurs négociées des paramètres:
 l'expéditeur annonce les valeurs de LQA_{max}, CHQ_{min} et OT aux destinataires via des paquets HB et JC.

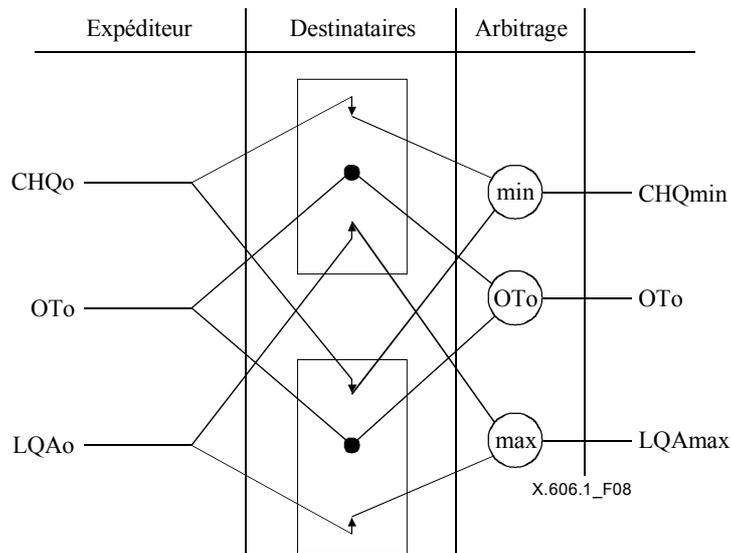


Figure 8 – Procédures de négociation de la QS

Pour ce qui est des paramètres de temps de transmission, de gigue et de taux de perte, on obtiendra une valeur LQA_{min} et non pas une valeur LQA_{max}, étant donné que les valeurs CHQ ne sont pas utilisées et que valeur de OT < valeur de LQA pour ces paramètres. C'est-à-dire que:

$$LQA_{min} = \min LQA_i, \text{ pour chaque destinataire } i.$$

La négociation de QS n'est pas effectuée pour les destinataires tardifs. L'expéditeur notifie simplement les valeurs négociées des paramètres aux destinataires tardifs via un paquet JC.

8.1.2 Négociation de la QS dans une hiérarchie arborescente

Si l'on utilise une arborescence de commande à plus de deux niveaux, chaque propriétaire LO parent doit exécuter des procédures d'arbitrage de QS pour les valeurs modifiées proposées par ses enfants (étape 3 du § 8.1.1). Dans ce cas, la règle d'arbitrage est la suivante:

$$CHQ_{min}' = \min CHQ_i, \text{ pour chaque enfant } i$$

$$LQA_{max}' = \max LQA_i, \text{ pour chaque enfant } i$$

Le parent fournit les valeurs choisies, CHQ_{min}' et LQA_{max}', à son parent via un paquet CC. Ainsi, l'expéditeur reçoit des paquets CC de tous ses enfants et obtient alors les valeurs choisies des paramètres pour tous les destinataires.

8.1.3 Négociation MSS

La taille MSS représente la taille maximale des paquets, qui dépend de l'unité maximale de transmission (MTU, maximum transmission unit) pour les transmissions de couche Liaison. Les valeurs types de MTU sont de 1500 octets pour Ethernet, 1492 octets pour l'IEEE 802, 4352 octets pour le FDDI, 576 octets pour la X.25, etc. L'unité MTU détermine la taille de trame pour les transmissions de couche Liaison et par conséquent la valeur de MSS.

La valeur par défaut de la taille MSS du protocole ECTP est de 1024 octets, et est compatible avec la plupart des types de liaison sauf les liaisons X.25. Si l'expéditeur et/ou le destinataire ne peut pas identifier sa valeur de MSS, il prend la valeur du MSS par défaut, à savoir 1024 octets.

Dans le protocole ECTP, si la négociation de la taille *MSS* doit être effectuée (voir § 7.3), les procédures suivantes sont exécutées:

- 1) l'expéditeur indique sa valeur de la taille *MSS* dans le paquet CR, et la transmet à tous les destinataires;
- 2) chaque destinataire propose sa propre valeur de la taille *MSS* via le paquet CC. La valeur de la *MSS* sera affectée par l'unité *MTU* du réseau local dont relève le destinataire. Si la valeur de la *MSS* du destinataire est supérieure à celle de l'expéditeur, le destinataire prend la valeur de la *MSS* de l'expéditeur;
- 3) l'expéditeur procède à l'arbitrage de la longueur *MSS* de la connexion en prenant la valeur minimale de la taille *MSS* pour tous les destinataires.

8.1.4 Réserve des ressources

Le protocole ECTP ne peut pas lui-même garantir les niveaux de QS exigés par les applications. Toutefois, les valeurs cibles des paramètres de QS peuvent être utilisées lors de la réserve des ressources de réseaux tels des services intégrés avec protocole RSVP et services différenciés (Diffserv).

Le modèle de protocole RSVP permet une bonne correspondance avec le protocole ECTP, étant donné que la réserve des ressources est effectuée de bout en bout. Dans les réseaux où le protocole RSVP est activé, des messages périodiques PATH RSVP seront remis aux destinataires. En réponse au message PATH, chaque destinataire envoie des messages périodiques RESV RSVP à l'expéditeur par un conduit de données multidiffusées.

Si le protocole RSVP est utilisé avec le protocole ECTP, les valeurs cibles des paramètres de QS de l'ECTP seront indiquées dans la configuration des descripteurs de trafic RSVP pour le TSPEC (ou FLOWSPEC) RSVP contenu dans le message PATH RSVP. Ces messages PATH seront émis conformément au paquet CR et aux paquets HB transmis par l'expéditeur. A noter que les modules RSVP sont gérés de manière distincte du protocole ECTP dans les systèmes d'extrémité. Cela signifie qu'une réserve de ressources et les fonctions de commande du protocole RSVP seront effectuées seulement par les démons (daemons) RSVP.

L'Annexe A contient un exemple d'interfonctionnement entre les protocoles ECTP et RSVP, ainsi qu'un schéma de mappage possible entre les paramètres de QS et de trafic associés.

Le modèle de service différencié définit des classes de service différencié pour le trafic IP, permettant de prendre en charge différents types d'applications et des besoins d'activités spécifiques. Une courte séquence binaire placée dans chaque paquet, dans l'octet TOS (ou DSCP) IPv4 ou dans l'octet de classe de trafic IPv6, est utilisée pour "marquer" un paquet de manière à ce qu'il reçoive un traitement subséquent particulier, ou comportement par sauts, dans chaque nœud de réseau. Il est nécessaire d'avoir une compréhension commune de l'utilisation et de l'interprétation de cette séquence binaire pour en avoir une utilisation interdomaniale, une interopérabilité multifournisseurs et un raisonnement cohérent sur les comportements de service attendus dans un réseau. Jusqu'à présent, on n'a pas trouvé de schéma d'interfonctionnement spécifique entre le protocole ECTP et Diffserv.

Le protocole ECTP devra utiliser des mécanismes de QS de réseaux sous-jacents, tel le RSVP ou le Diffserv pour établir des connexions de réseau offrant les niveaux de QS requis. Toutefois, l'utilisation potentielle du protocole ECTP n'est pas limitée aux méthodes actuelles applicables à la QS des réseaux. Le protocole ECTP est conçu pour permettre d'améliorer la QS avec les divers modèles de QS qui pourraient être élaborés à l'avenir.

8.2 Surveillance de la QS

La fonction de surveillance de la QS permet à l'expéditeur de disposer d'informations sur le fonctionnement de la connexion. Pour cela, chaque destinataire est tenu de mesurer les valeurs des paramètres constatés et de les signaler en retour à l'expéditeur.

En ce qui concerne les paramètres de QS utilisés dans la connexion, chaque destinataire mesure les valeurs des paramètres qu'il observe. La valeur mesurée est mappée en valeur d'état de paramètre pour chaque paramètre. Un état de paramètre est un entier dont la valeur est 0, 1, 2 ou 3. Cette valeur d'état est inscrite dans l'élément accusé de réception et acheminée à l'expéditeur au moyen d'un paquet ACK subséquent (voir § 7.4). L'expéditeur compile les valeurs d'état de paramètre en provenance de tous les destinataires.

L'objet de la surveillance de QS est de donner à l'expéditeur des informations sur l'état de QS de la connexion. Sur la base de l'information d'état surveillée, l'expéditeur peut également prendre, au besoin, des mesures de maintien de la QS.

8.2.1 Génération de l'ACK

Chaque destinataire signale les *valeurs d'état de paramètre* à son parent en générant des paquets ACK. Dans le protocole ECTP-2, la règle de génération des paquets ACK est quelque peu différente de celle spécifiée pour l'ECTP-1. Dans l'ECTP-2, on pose l'*AGT* égal à l'*AGN*. Plus particulièrement, l'*AGT* est rendu égal à "*AGN* x 1 seconde, c'est-à-dire, *AGN* secondes". Chaque destinataire générera donc des paquets ACK périodiques chaque seconde *AGT*. Le réglage de l'*AGN* dépend de l'implémentation, mais l'*AGN* est mis à "8" par défaut dans l'ECTP-2.

Pour générer un paquet ACK, chaque destinataire doit disposer d'un temporisateur, à savoir un chronomètre de surveillance de la QS (*QMT*, *QoS monitoring time*), en secondes. Le chronomètre *QMT* est activé dès que le destinataire termine l'établissement de la connexion, c'est-à-dire après la réception d'un paquet CC ou JC en provenance de l'expéditeur. Le chronomètre *QMT* décompte le temps de manière monotone pendant toute la durée de la connexion, c'est-à-dire qu'il n'y a pas de rafraîchissement au cours de la connexion.

Chaque destinataire transmet un paquet ACK à son parent si la relation suivante est vérifiée:

$$QMT \% AGN = ID \text{ Enfant } \% AGN$$

Ce schéma est employé pour minimiser les implosions d'ACK du côté parent dans la mesure du possible. Par ce mécanisme, chaque destinataire continuera à générer des paquets ACK chaque *AGN* seconde, après démarrage du chronomètre *QMT*. Par exemple, si *AGN* est égal à 8, le destinataire avec l'*ID enfant* de 3 ou 11 générera des paquets ACK aux instants *QMT* 3, 11, 19, 27 secondes, etc. A noter dans cet exemple que le premier paquet ACK sera généré pour des paquets de données transmises seulement pendant 3 secondes et non 8 secondes. Toutefois, les autres paquets ACK suivants seront générés pour un intervalle de temps de *AGN* secondes. De cette manière, chaque destinataire générera des paquets ACK toutes les *AGN* secondes, à l'exception du premier paquet ACK.

Chaque paquet ACK achemine la valeur d'état de chaque paramètre mesuré tel que décrit au § 8.2.2.

8.2.2 Mesure des valeurs des paramètres de QS

Pour la transmission de données multidiffusée, chaque destinataire mesure les valeurs observées de chacun des paramètres de QS. Toutes les valeurs des paramètres sont mesurées, enregistrées et calculées, jusqu'à ce qu'un nouveau paquet ACK soit généré conformément à la règle de génération applicable. Lorsque le moment vient d'envoyer un paquet ACK, le destinataire calcule la valeur d'état du paramètre pour les paquets de données reçus et collectés jusqu'à ce moment. Après la transmission d'un paquet ACK, les données collectées sont effacées et de nouvelles données seront collectées et enregistrées pour générer un nouveau paquet ACK.

8.2.2.1 Débit

Le débit mesuré est le débit de réception des données en octets par seconde. Le débit de réception des données est comme étant égal au:

$$\text{volum e de paquets de données reçus en octets pendant } AGN \text{ secondes.}$$

Une fois encore, le premier paquet ACK peut être généré avant que *AGN* secondes se soient écoulées. Pour mesurer la valeur du débit, un récepteur doit conserver les informations sur le nombre de paquets de données (octets) qui ont été reçus pendant une durée spécifiée.

Chaque fois qu'une nouvelle valeur de débit est obtenue, cette valeur est mappée en une *valeur d'état de paramètre*, c'est-à-dire en un entier égal à 0, 1, 2 ou 3 selon la règle de mappage décrite au § 8.2.3.

8.2.2.2 Taux de perte de données

Le taux de perte de données représente le taux de perte de paquets et est exprimé en pourcentage. Le taux de perte de paquets est déterminé comme étant égal au:

$$\text{nombre de paquets perdus sur nombre de paquets de données reçus pendant } AGN \text{ secondes.}$$

Une fois encore, le premier paquet ACK peut être généré avant que *AGN* secondes aient été écoulées. Pour mesurer la valeur du taux de perte de données, le récepteur doit conserver l'information sur le nombre de paquets de données qui ont été perdus. Lorsqu'un récepteur génère un paquet ACK, la valeur du taux de perte alors mesurée est mappée en une *valeur d'état de paramètre*.

8.2.2.3 Temps de transit et gigue de temps de transit

Pour mesurer le temps de transit de bout en bout, l'expéditeur doit transmettre des paquets de données avec un élément d'horodatage. Il faut aussi que les horloges de l'expéditeur et des récepteurs soient synchronisées. Sans le respect de ces deux conditions, il est difficile d'obtenir une information exacte sur le temps de transit et la gigue associée. Dans le présent paragraphe, on suppose que chaque destinataire peut mesurer le temps de transit de bout en bout d'un paquet émis par l'expéditeur.

Le temps de transit est mesuré pour chaque paquet de données reçu. Ces valeurs de temps de transit sont moyennées sur le nombre de paquets de données reçus pendant *AGN* secondes. La gigue de temps de transit est mesurée comme étant la différence entre les valeurs maximales et minimales des temps de transit des paquets de données reçus.

Pendant les *AGN* secondes, chaque fois qu'un nouveau paquet de données arrive, le temps de transit est calculé, et les moyennes de temps de transit et de gigue sont actualisées. Juste avant qu'un destinataire génère un paquet ACK, la valeur courante mesurée est mappée en une *valeur d'état de paramètre*.

8.2.3 Mappage en une valeur d'état d'un paramètre

La valeur de paramètre mesurée est mappée en une valeur d'état de paramètre pour chaque paramètre de QS. Le paquet ACK subséquent contiendra la valeur d'état. A noter de nouveau que la mesure d'un paramètre de QS n'est activée que si l'utilisation du paramètre dans la connexion est indiquée (voir § 7.3).

La Figure 9 illustre le mappage d'une valeur mesurée en une valeur d'état.

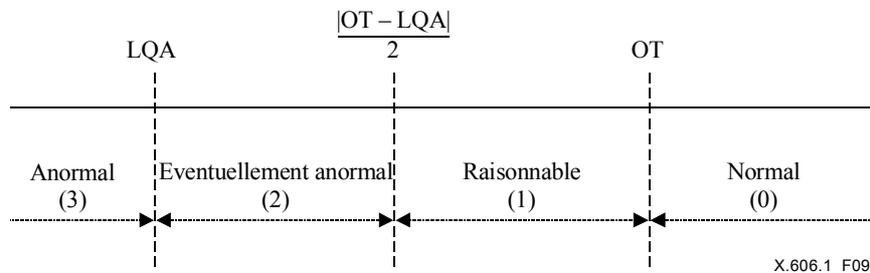


Figure 9 – Mappage d'une valeur mesurée en une valeur d'état

Comme le montre la figure, le mappage d'une valeur de paramètre mesurée en une valeur d'état est fondé sur les valeurs OT et LQA des paramètres. Le protocole ECTP utilise également une valeur "seuil" pour classer l'état en normal (0), raisonnable (1), éventuellement anormal (2) et anormal (3). La valeur de *seuil* est choisie comme étant la valeur médiane entre des valeurs de paramètre OT et LQA (c'est-à-dire $|LQA - OT| / 2$) comme indiqué dans la figure.

Une *valeur initiale d'état de paramètre* est mise à "0". Après mesure de sa valeur, le mappage en une *valeur d'état de paramètre* est effectué comme suit:

SI "*valeur de paramètre mesurée* > OT", état = 0,

SI "*seuil* < *valeur de paramètre mesurée* ≤ OT", état = 1,

SI "*LQA* < *valeur de paramètre mesurée* ≤ *seuil*", état = 2,

SI "*valeur mesurée* ≤ *LQA*", état = 3.

Dans les règles de mappage ci-dessus, les inégalités ne sont valables que pour le paramètre débit. Pour les autres paramètres tels le temps de propagation, la gigue et le taux de perte, ces égalités doivent être inversées car les valeurs OT sont inférieures ou égales aux valeurs LQA.

8.2.4 Rapport à l'expéditeur

Chaque destinataire signale les *valeurs d'état de paramètre* obtenues à son parent via des paquets ACK. Les paquets ACK sont générés sur la base de l'*ID enfant* et de *AGN* secondes. Ainsi, toutes les informations concernant les *valeurs d'état de paramètre* sont remises vers l'expéditeur en respectant la hiérarchie de l'arborescence.

8.2.4.1 Compilation des paquets ACK par le propriétaire LO dans une hiérarchie arborescente

Dans une hiérarchie arborescente, chaque propriétaire LO parent compile les paquets ACK provenant de ses enfants. Cette compilation se fera juste avant que le parent génère son propre paquet ACK. Le propriétaire LO génère son propre paquet ACK toutes les *AGN* secondes, de même que les destinataires. Le parent prend tout simplement une valeur moyenne des valeurs d'état de paramètre signalées par son enfant ainsi que sa propre valeur mesurée, juste avant de générer les paquets ACK.

La valeur moyenne est calculée pour chaque paramètre de la manière suivante:

somme des valeurs d'état de paramètre provenant des enfants répondeurs sur le CCN.

Dans une structure arborescente à plusieurs niveaux, le nombre de descendants est représenté par le numéro de destinataire actif (*ARN*, *active receiver number*) enregistré dans l'élément membre de l'arborescence. Dans ce cas, les valeurs d'état des paramètres signalées par les enfants sont pondérées par leur valeur *ARN* comme suit:

$$\text{valeur moyenne pondérée} = \frac{\sum_{i=1, \dots, \text{CCN}} \{\text{ARN}(i) \times \text{valeur d'état de paramètre}(i)\}}{\sum_{i=1, \dots, \text{CCN}} \text{ARN}(i)}$$

Un propriétaire LO parent arrondit cette valeur moyenne à 0, 1, 2 ou 3 pour chaque paramètre de QS et construit son paquet ACK.

8.2.4.2 Compilation par l'expéditeur

La compilation des valeurs d'état des paramètres par l'expéditeur est la même que pour le propriétaire LO. L'expéditeur effectue également une compilation des paquets ACK signalés par ses enfants toutes les *AGN* secondes.

Après compilation des paquets ACK, l'expéditeur prend simplement la valeur moyenne pondérée par le nombre de destinataires actifs (*ARN*) de chaque enfant. A savoir:

valeur d'état agrégée = somme pondérée des valeurs d'état signalées sur le nombre ARN de la connexion.

L'expéditeur obtient une valeur d'état compilée pour chaque paramètre de QS. Plus particulièrement, l'expéditeur aura les valeurs d'état compilées suivantes (si chacun des paramètres est utilisé dans la connexion):

- a) état compilé du débit, désigné par *Tvalue*;
- b) état compilé du temps de transit, désigné par *Dvalue*;
- c) état compilé de la gigue du temps de transit, désigné par *Jvalue*;
- d) état compilé du taux de perte de données, désigné par *Lvalue*.

Les valeurs compilées sont comprises entre 0 et 3 inclus.

L'expéditeur peut faire suivre l'information d'état surveillée vers l'application. L'information surveillée est utile pour faire un diagnostic du fonctionnement d'une connexion en termes de QS, qui peut être ensuite utile pour concevoir un modèle de facturation/taxation.

L'information obtenue par surveillance est également utilisée pour le maintien de la QS. Parmi les valeurs faisant l'objet d'une surveillance, *Lvalue* est utilisée pour ajuster le débit de transmission de données (voir § 8.3.1). Une valeur de somme pondérée pour toutes les valeurs d'état, *Tvalue*, *Dvalue*, *Jvalue* et *Lvalue* peut être utilisée pour déclencher une pause dans la connexion, et sa terminaison.

8.3 Maintien de la QS

Le maintien de la QS est exécuté pour maintenir la qualité d'une connexion au niveau désiré et éviter que sa qualité ne devienne inférieure à celle du niveau de QS négocié.

Sur la base des valeurs d'état des paramètres surveillées, l'expéditeur peut prendre les actions de maintien de QS suivantes:

- 1) ajustement du débit de transmission de données;
- 2) pause et reprise de la connexion;
- 3) élimination de l'élément perturbateur;
- 4) fin de la connexion.

L'ajustement du débit de transmission est lié à la gestion du flux et des encombrements fondés sur le débit. La pause et la reprise d'une connexion ainsi que sa terminaison sont des actions qui peuvent être prises pour gérer la connexion. Ces événements seront annoncés à tous les destinataires via des paquets ND et CT émis par l'expéditeur.

Pour déclencher ces actions de maintien de la QS, l'expéditeur doit configurer les valeurs seuil suivantes:

- 1) *threshold_rate_increase* (augmentation du débit seuil) et *threshold_rate_decrease* (diminution du débit seuil) pour ajuster le débit de transmission de données;
- 2) *threshold_connection_pause*.

Toutes les valeurs seuil sont des nombres réels allant de 0 à 3 inclus.

8.3.1 Ajustement du débit de transmission de données

Le protocole ECTP utilise une commande de flux fondée sur une fenêtre de taille fixe. La taille de fenêtre par défaut est la même que la taille *Bitmap ACK (ABS)*, 32. L'expéditeur peut au maximum transmettre des paquets de données *taille de fenêtre* au débit *DTR (data transmission rate)*. Le protocole ECTP assure la gestion des encombrements en réglant de manière dynamique le débit *DTR*, sur la base des valeurs d'état du taux de perte *Lvalue* (voir § 8.2.4.2).

L'ajustement du débit de transmission de données est fondé sur les *threshold_rate_increase* et *threshold_rate_decrease*, qui sont préfigurées par l'expéditeur à partir des caractéristiques de l'application. Ces valeurs doivent respecter les conditions suivantes:

$$0 \leq \textit{threshold_rate_increase} \leq \textit{threshold_rate_decrease} \leq 3$$

Les valeurs par défaut sont *threshold_rate_increase* = 1.0 et *threshold_rate_decrease* = 2.0.

Dans la phase de transmission de données, l'expéditeur utilise au début un débit *DTR = LQA*, et le débit *DTR* peut être ajusté comme suit:

$$\text{débit } LQA \leq DTR \leq \text{débit } CHQ.$$

Chaque AGN secondes, l'expéditeur ajuste le débit *DTR* sur la base de *threshold_rate_increase*, *threshold_rate_decrease* et la valeur *Lvalue* surveillée comme suit:

```

SI      Lvalue < threshold_rate_increase
ALORS  DTR = Min {CHQ, DTR + Transmission Rate Increase (TRI)}
SI      threshold_rate_increase ≤ Lvalue ≤ threshold_rate_decrease
ALORS  DTR n'est pas modifié
SI      Lvalue > threshold_rate_decrease
ALORS  DTR = Max {LQA, DTR – Transmission Rate Decrease (TRD)}
```

Les variables d'ajustement de débit tels *TRI* et *TRD* peuvent être fondées sur les débits *CHQ* et *LQA*. Par exemple:

$$TRI = (CHQ - LQA) \times 1/20$$

$$TRD = (CHQ - LQA) \times 1/5$$

8.3.2 Pause et reprise de la connexion

La pause de connexion peut être effectuée par l'expéditeur pour suspendre temporairement la transmission de données multidiffusées afin d'empêcher une forte dégradation de la qualité de la connexion.

Une pause et une reprise de la connexion peuvent être effectuées, en fonction des exigences de l'application. Dans ce cas, l'application expéditrice déclenchera la pause de la connexion sur la base des valeurs d'état des paramètres surveillés telles *Tvalue*, *Dvalue*, *Jvalue* et *Lvalue*. Si la pause de la connexion est déclenchée, l'expéditeur peut envoyer des paquets ND périodiques dont le bit *F* est mis à "1" dans l'en-tête fixe (voir la Rec. UIT-T X.606 | ISO/CEI 14476-1). L'expéditeur peut ne pas transmettre de nouveaux paquets DT, tandis que les paquets de commande y compris les paquets HB peuvent être envoyés. Chaque destinataire peut également envoyer des paquets de contrôle tels des paquets ACK.

La pause de la connexion peut également être déclenchée, sur la base d'un seuil préconfiguré *threshold_connection_pause*. Dans ce cas, seulement si la valeur de l'état de la connexion surveillée est supérieure au seuil *threshold_connection_pause*, l'expéditeur ECTP déclenche une pause de la connexion. La valeur suggérée du seuil *threshold_connection_pause* est 2,5.

Pour cela, la valeur d'état de connexion est calculée pour toutes les valeurs d'état des paramètres surveillés comme suit:

$$\textit{Etat de la connexion} = Tweight \times Tvalue + Dweight \times Dvalue + Jweight \times Jvalue + Lweight \times Lvalue.$$

Chacune des valeurs pondérées doit être également configurée ainsi que *threshold_connection_pause*, les contraintes suivantes étant imposées:

$$0 \leq Tweight, Dweight, Jweight, Lweight \leq 1$$

$$Tweight + Dweight + Jweight + Lweight = 1$$

Une valeur pondérée est mise à "0" si le paramètre de QS correspondant n'est pas activé pour la connexion.

La pause de connexion est déclenchée si:

$$Etat\ de\ la\ connexion \geq threshold_connection_pause$$

Après qu'une pause de la connexion ait été indiquée, si la durée de pause de la connexion (CPT, *connection pause time*) est écoulée, une reprise de la connexion est déclenchée et l'expéditeur commence à transmettre des données multidiffusées avec le débit de transmission *LQA*. Lorsqu'il y a indication de la reprise de la connexion, des paquets ND positionneront le bit *F* de l'en-tête à "0".

8.3.3 Elimination de l'élément perturbateur

L'expéditeur ou le propriétaire LO peut demander l'élimination de l'élément perturbateur afin de maintenir l'état de la QS à un niveau souhaité et également empêcher une dégradation encore plus importante de l'état de la connexion. Le schéma détaillé de l'élimination de l'élément perturbateur peut dépendre de l'implémentation et être fondé sur les valeurs d'état des paramètres définies dans la présente Spécification.

Par exemple, un destinataire peut être éliminé par son parent s'il lui est associée une valeur d'état de paramètre plus grande que *threshold_connection_pause* et supérieure de plusieurs fois à un seuil préconfiguré. Le schéma d'élimination de l'élément perturbateur doit être conçu et appliqué avec soin, étant donné que l'opération d'élimination peut avoir un effet important sur le comportement global du protocole ECTP.

8.3.4 Terminaison d'une connexion

L'option naturelle pour la terminaison d'une connexion consiste à mettre fin à une connexion après que toutes les données multidiffusées ont été transmises. Dans les opérations de gestion de la QS, la terminaison d'une connexion est également déclenchée si l'état de la connexion est perçu comme étant "irré récupérable".

La terminaison d'une connexion doit être effectuée conformément aux besoins de l'application. Si une terminaison de la connexion est déclenchée, l'expéditeur émet un paquet CT à l'intention de tous les destinataires, et met fin à la connexion.

La terminaison d'une connexion peut également être déclenchée, sur la base d'un temps de terminaison de connexion (CTT, *connection termination time*) préconfiguré. Dans ce cas, la terminaison d'une connexion est déclenchée si:

une pause de connexion subséquente se produit de nouveau à l'intérieur du temps *CTT*
à partir de la reprise de la connexion.

Le temporisateur *CTT* est activé lorsque la reprise de la connexion est indiquée. La terminaison d'une connexion peut ne pas être prise en charge par certaines applications.

9 Temporisations et variables

Les temporisateurs et les variables suivants sont utilisés pour la gestion de la QS.

9.1 Temporisations

- a) Intervalle de génération de paquets ACK (*AGT*, *ACK generation time*) en secondes: chaque destinataire génère un paquet ACK périodiquement toutes les *AGT* secondes, sauf pour le premier paquet ACK. Dans l'ECTP-2, le temps *AGT* est égal à *AGN* secondes (voir § 8.2).
- b) Durée de pause de la connexion (CPT, *connection pause time*) en secondes: durée d'une pause appliquée à une connexion (voir § 8.3.2).
- c) Intervalle de temps de terminaison de connexion (CTT, *connection termination time*) en secondes: lorsque, pendant l'intervalle de temps *CTT*, la reprise d'une connexion se traduit par une nouvelle pause de la connexion, il est mis fin à cette connexion (voir § 8.3.4).

- d) Durée de surveillance de la QS (QMT, *QoS monitoring time*) en secondes: chaque destinataire mesure les valeurs des paramètres de QS toutes les *QMT* secondes (voir § 8.2.2).

9.2 Variables opérationnelles

- a) Valeurs compilées des états de paramètre: l'expéditeur procède à la compilation des valeurs d'état de paramètre signalées par les destinataires et obtient ainsi les valeurs *Tvalue*, *Dvalue*, *Jvalue* et *Lvalue* (voir § 8.2).
- b) Nombre de générations de paquets ACK (*AGN*, *ACK generation number*): chaque destinataire génère un paquet ACK périodiquement toutes les *AGT* secondes. Dans l'ECTP-2, l'*AGT* est égal à *AGN* secondes (voir § 8.2). Par défaut, la valeur *AGN* est égale à 8.
- c) Etat de la connexion: pour déclencher une pause/reprise de connexion, toutes les valeurs d'état de paramètre compilées peuvent être pondérées moyennées avec les pondérations de paramètres préconfigurées telles *Tweight*, *Dweight*, *Jweight* et *Lweight*. On obtient ainsi l'*état de la connexion* qui représente l'état global de la connexion (voir § 8.3.2).
- d) Débit de transmission de données (*DTR*, *data transmission rate*): l'expéditeur transmet les données multidiffusées avec le débit *DTR* (voir § 8.3.1).
- e) Valeur mesurée des paramètres: chaque destinataire est tenu de mesurer la valeur observée des paramètres de QS utilisés dans la connexion. On obtient ainsi la *valeur mesurée des paramètres* (voir § 8.2).
- f) Valeur d'état de paramètre: la *valeur mesurée des paramètres* est mappée en *valeur d'état des paramètres*, qui est un entier égal à 0, 1, 2 ou 3 (voir § 8.2).
- g) Diminution du débit de transmission (*TRD*, *transmission rate decrease*): la valeur du débit *DTR* est diminuée de *TRD* (voir § 8.3.1).
- h) Augmentation du débit de transmission (*TRI*, *transmission rate increase*): le débit *DTR* est augmenté de *TRI* (voir § 8.3.1).
- i) *Threshold_rate_increase* (*seuil d'augmentation du débit*): seuil à partir duquel on augmente le débit de transmission de données (voir § 8.3.1).
- j) *Threshold_rate_decrease* (*seuil de diminution du débit*): seuil à partir duquel on diminue le débit de transmission de données (voir § 8.3.1).
- k) *Threshold_connection_pause* (*seuil de déclenchement d'une pause de connexion*): seuil à partir duquel on déclenche une pause de la connexion (voir § 8.3.2).

Annexe A

Interfonctionnement entre les protocoles ECTP et RSVP pour la réservation des ressources

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

La QS avec le protocole ECTP peut être garantie par interfonctionnement avec le RSVP/Intserv pour les protocoles de réservation des ressources de réseau. Sur la base des valeurs des paramètres de QS, le protocole RSVP peut configurer ses descripteurs de trafic s'il est activé dans le réseau.

La présente annexe décrit comment utiliser la signalisation RSVP en association avec le protocole ECTP pour réserver des ressources et comment mapper les paramètres de QS de l'ECTP en descripteurs de trafic RSVP. Tous les schémas illustrés ci-dessous présupposent le mode sans négociation (voir § 7.1 et 8.1), dans lequel les valeurs des paramètres de QS proposées par l'expéditeur sont imposées à tous les destinataires. La façon d'utiliser le protocole RSVP dans le mode avec négociation appelle un complément d'étude. Toutefois, même dans le mode avec négociation, la relation de mappage des paramètres entre le protocole RSVP et le protocole ECTP reste valable. Dans ce cas, chacun des paramètres de QS négocié sera mappé en descripteurs de trafic RSVP.

A.1 Paramètres de QS ECTP

Les paramètres de QS de l'ECTP et leurs valeurs cibles sont résumés comme suit:

- a) débit: CHQ, OT et LQA;
- b) temps de transit: OT et LQA;
- c) gigue de temps de transit: OT et LQA;
- d) taux de perte de données: OT et LQA.

En conclusion, au moins du point de vue de la technologie RSVP actuelle, le temps de transit, la gigue de temps de transit et le taux de perte ne peuvent pas être explicitement pris en charge par le protocole RSVP. La signalisation RSVP garantit strictement le respect des exigences en matière de temps de propagation au niveau du réseau grâce au recours à la réservation de largeur de bande et à un séquenceur de traitement du trafic au niveau des routeurs intermédiaires sur le trajet. Le terme "temps de propagation" défini pour le protocole RSVP (délai de mise en file d'attente au niveau du réseau) est assez différent de celui défini pour le protocole ECTP (temps de transit de bout en bout). Le paragraphe ci-après donne un bref résumé des opérations associées au protocole RSVP.

A.2 Aperçu général du protocole RSVP

Le protocole RSVP (IETF RFC 2205, 2210, 2212) inclut les objets RSVP suivants:

- a) RSVP SENDER_TSPEC;
- b) RSVP ADSPEC;
- c) RSVP RECEIVER_FLOWSPEC.

A.2.1 Objet RSVP SENDER_TSPEC

La spécification du trafic (TSPEC) du protocole RSVP est un objet qui achemine les descripteurs de trafic générés par un expéditeur dans une session RSVP. Cet objet inclut les descripteurs de trafic suivants:

- a) la spécification du conteneur de jetons (token bucket) r et b , dans laquelle r est le taux de génération ou de fuite des jetons, et b la profondeur du conteneur;
- b) débit crête p ;
- c) unité surveillée maximale m ;
- d) taille de paquet maximale M .

Les descripteurs r et p sont mesurés en octets par seconde et b , m et M en octets. Les débits r et p sont les débits moyens et maximum du flux d'informations respectivement et b est un paramètre qui limite la variabilité de l'émission de trafic.

Les autres paramètres définissent des limites relatives à la distribution de longueur des paquets dans le flux. M est la taille maximale d'un paquet conforme et m est tel qu'un paquet dont la taille est inférieure à m est traité par l'entité de surveillance du réseau comme si sa taille était égale à m . Les tailles m et M ne sont pas directement concernées par le protocole ECTP.

A.2.2 Objet RSVP ADSPEC

La spécification d'annonce (ADSPEC) du protocole RSVP est un objet qui achemine l'information générée au niveau des expéditeurs de données ou au niveau d'éléments de réseaux intermédiaires. L'objet ADSPEC du RSVP circule dans le sens aval vers les destinataires et peut être utilisé et mis à jour dans le réseau avant d'être remis aux applications destinataires. Cette information inclut des paramètres décrivant les propriétés du trajet de données, y compris la disponibilité de services spécifiques de contrôle de la QS et des paramètres spécifiques des services de contrôle de la QS pour un fonctionnement correct. Cet objet inclut des descripteurs de trafic suivants:

- a) largeur de bande de la liaison avec la largeur de bande minimale disponible sur le trajet de bout en bout;
- b) temps de transmission de bout en bout à l'exception du délai de file d'attente; ces temps peuvent être mesurés par deux termes d'erreur à savoir *C* et *D*. Le terme d'erreur *C* dépend du débit, il représente le retard auquel un paquet du flux peut être soumis en raison des paramètres de débit du flux. Le terme d'erreur *D* ne dépend pas du débit, il s'agit d'un terme d'erreur par élément. Il représente, au moyen de l'élément de service, la variation de temps de transit non fondée sur le débit et correspondant au cas le plus défavorable. Il est en général calculé ou bien sa valeur est mise égale au temps d'amorçage ou de configuration.

A.2.3 Objet RSVP FLOWSPEC

Cet objet achemine l'information de demande de réservation générée par les destinataires et se compose des éléments RECEIVER_TSPEC et RECEIVER_RSPEC. L'information contenue dans cet objet circule dans le sens amont des destinataires de données aux sources de données. Il peut être utilisé ou mis à jour au niveau des éléments intermédiaires du réseau avant de parvenir à l'application émettrice.

L'élément RECEIVER_TSPEC est identique à l'élément SENDER_TSPEC sauf en ce qui concerne l'unité MTU du trajet de bout en bout. Par ailleurs, l'élément RECEIVER_RSPEC inclut les paramètres suivants:

- a) *R*: largeur de bande réservée;
- b) *S*: un terme de marge.

R doit être supérieur ou égal à *r* de l'objet TSPEC. Le terme marge est exprimé en millisecondes. Par marge, on entend la différence entre le temps de propagation souhaité et le temps de propagation obtenu en utilisant un niveau de réservation *R*. Ce terme peut être utilisé par l'élément de réseau pour diminuer les ressources réservées pour traiter le flux associé.

A.2.4 Interface API RSVP

La figure A.1 illustre la structure d'implémentation du protocole RSVP dans un système.

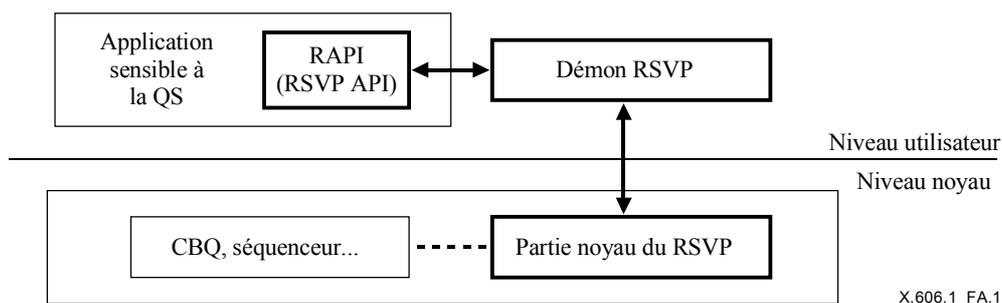


Figure A.1 – Structure d'implémentation du protocole RSVP

Dans un système, le protocole RSVP fonctionne avec trois modules: le noyau RSVP, le démon RSVP et le protocole RSVP (interface RAPI). Le noyau RSVP gère la mise en file d'attente et le séquençement des données conformément aux exigences associées du protocole RSVP de couche supérieure. Le démon RSVP gère la signalisation entre l'interface RAPI et le noyau RSVP. L'interface RAPI assure l'interface avec les applications sensibles à la QS et qui fait en sorte que les opérations RSVP prennent en charge les exigences de QS émanant des applications. Les caractéristiques de QS du protocole ECTP seront intégrées et appliquées aux réseaux compatibles RSVP via l'interface RAPI.

Les fonctions de l'interface RAPI peuvent être classées en deux groupes: des fonctions appelées par une application et les utilitaires asynchrones (appel ascendant) remis à une application par le démon RASP.

Ces fonctions de base du protocole RSVP sont les suivantes:

- `rapi_session()`, qui est utilisé pour déclencher une session RSVP;

- `rapi_sender()`, qui est utilisé par un expéditeur pour envoyer des messages PATH RSVP;
- `rapi_reserve()`, qui est utilisé par un destinataire pour réserver des ressources réseau;
- `rapi_release()`, qui est utilisé pour mettre fin à une session RSVP;
- `rapi_getfd()` et `rapi_dispatch()`, qui sont utilisés par une application pour obtenir des informations d'état concernant le protocole RSVP via les utilitaires d'appel ascendant asynchrones.

A.3 Exemple de mappage des paramètres entre protocoles RSVP et ECTP

Le Tableau A.1 résume un exemple de mappage de paramètre entre protocoles RSVP et ECTP, qui peut être utilisé pour prendre en charge la QS dans le protocole RSVP.

NOTE – Un mappage différent peut également être utilisé selon l'implémentation.

Tableau A.1 – Mappage de paramètres entre protocoles ECTP et RSVP

Paramètres RSVP	Paramètres de QS ECTP	Description
<i>p</i> (TSPEC)	Débit CHQ	Mappé directement en <i>p</i>
<i>r</i> (TSPEC)	Débit OT	Mappé directement en <i>r</i>
<i>b</i> (TSPEC)	(débit CHQ – débit OT) × un temps arbitraire (1 ~ 3 secondes)	La taille du conteneur de jetons est fixée arbitrairement sur la base des paramètres de débit ECTP
<i>m</i> (TSPEC)	En-tête fixe IP + UDP + ECTP (20 + 8 + 16 = 44 octets)	Directement mappé
<i>M</i> (TSPEC)	MSS ECTP (1024 octets par défaut)	Directement mappé
<i>R</i> (RSPEC)	(Débit CHQ + débit OT)/2	La réservation est mise à la valeur moyenne des débits CHQ et OT
<i>S</i> (RSPEC)	0	Le terme marge n'est pas positionné

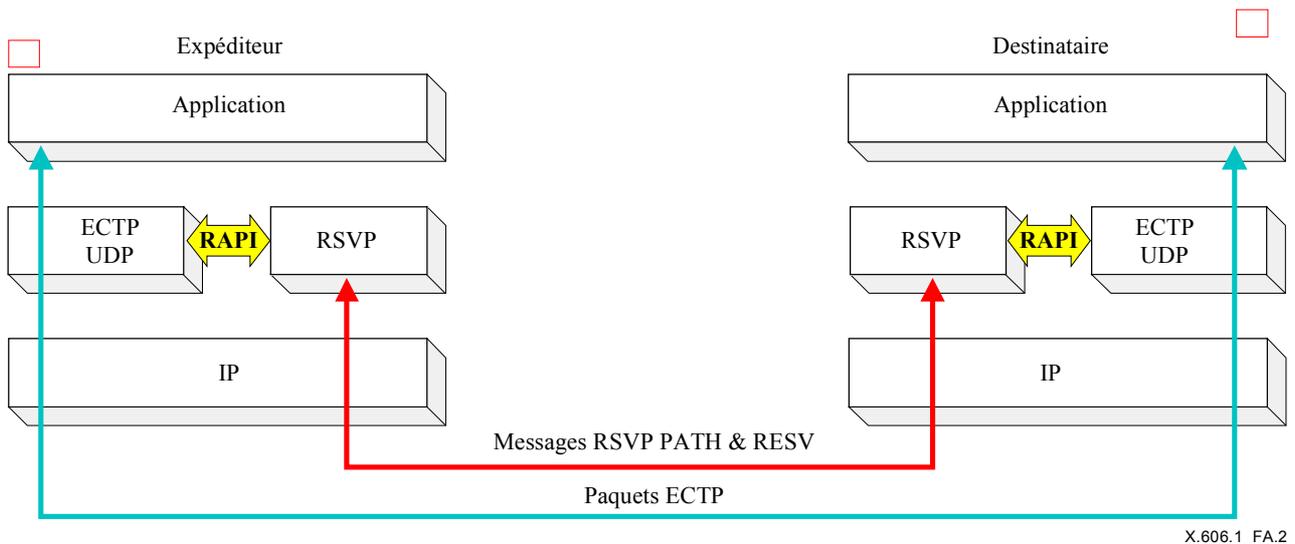
Lorsqu'un démon ECTP appelle un démon RSVP, les paramètres de QS ECTP sont mappés en descripteurs de trafic TSPEC RSVP tels *p*, *r*, *b*, *m*, *M*, comme le montre le tableau. Ces valeurs de paramètres TSPEC provenant de l'expéditeur sont remises aux destinataires via un message PATH RSVP.

Dès qu'un destinataire reçoit un message PATH RSVP, l'information contenue dans les objets SENDER_TSPEC et ADSPEC est transmise via l'interface API RSVP à l'application ECTP. L'application interprète l'information qui arrive et l'utilise pour choisir les paramètres de réservation des ressources. Ces paramètres sont placés dans un objet FLOWSPEC RSVP et sont transmis à l'expéditeur via les messages RESV RSVP. La réservation de largeur de bande *R* et le terme de marge *S* qui sont contenus dans l'objet RSPEC RSVP sont mappés en utilisant les paramètres ECTP comme le montre le Tableau A.1.

A.4 Scénario d'interfonctionnement entre protocoles ECTP et RSVP

Les valeurs cible de chacun des paramètres de QS ECTP peuvent être utilisées pour construire un objet SENDER_TSPEC RSVP. Pour cela, une interface entre les protocoles ECTP et RSVP est nécessaire afin que les valeurs de paramètres de QS ECTP soient communiquées au processeur RSVP. L'objet SENDER_TSPEC sera remis au destinataire via des messages PATH RSVP.

L'instant où le protocole ECTP appelle le protocole RSVP dépend de l'implémentation, mais on recommande que l'expéditeur ECTP n'invoque le protocole RSVP qu'après avoir envoyé un paquet CR et que le destinataire ne réagisse qu'après réception du paquet CR. La Figure A.2 illustre un modèle d'interfonctionnement entre protocoles ECTP et RSVP.



X.606.1_FA.2

Figure A.2 – Modèle d'interfonctionnement entre protocole ECTP et RSVP

Comme le montre la figure, les protocoles ECTP et RSVP communiquent entre eux via l'interface API RSVP (RAPI). L'expéditeur ECTP transfère les informations relatives aux paramètres de QS au protocole RSVP via l'interface RAPI, lorsqu'il invoque le démon RSVP. Après activation du protocole RSVP par le protocole ECTP, les communications RSVP se font entre le démon RSVP expéditeur et les démons RSVP destinataires. C'est-à-dire, une relation d'homologue à homologue RSVP est établie entre l'expéditeur et les différents destinataires.

Pendant la connexion ECTP, chaque module RSVP fait rapport de la situation en matière de réservation des ressources de réseau au démon ECTP associé via l'interface RAPI, ce qui permettra de vérifier que la réservation a bien eu lieu ou de constater qu'il y a une condition d'erreur. Voici un exemple d'ensemble de codes d'état RSVP:

- code d'état 0: les événements PATH sont indiqués (utilisés par le RSVP destinataire);
- code d'état 1: les événements d'erreur PATH sont indiqués (utilisés par le RSVP expéditeur);
- code d'état 2: les événements RESV sont indiqués (utilisés par le RSVP expéditeur);
- code d'état 3: les événements d'erreur RESV sont indiqués (utilisés par le RSVP destinataire);
- code d'état 4: les événements RESV_CONFIRM sont indiqués (utilisés par le RSVP destinataire).

Le noyau du protocole ECTP peut obtenir du démon RSVP ces codes d'état au moyen des fonctions d'appel ascendant asynchrones RSVP.

Les procédures exécutées par les modules ECTP et RSVP et destinées à l'interfonctionnement entre ces protocoles peuvent être résumées comme suit:

- (1) l'expéditeur ECTP transmet un paquet CR au destinataire. Le paquet CR contient un élément de QS indiquant les valeurs de paramètre de QS exigées par l'application;
- (2) l'expéditeur ECTP invoque son démon RSVP associé;
- (3) l'expéditeur RSVP élabore un message PATH qui contient les paramètres TSPEC fondés sur les paramètres de QS ECTP;
- (4) l'expéditeur RSVP transmet périodiquement les messages PATH aux destinataires;
- (5) en cas d'indication d'une erreur PATH RSVP, l'expéditeur RSVP communique le code d'état correspondant à l'expéditeur ECTP;
- (6) dès qu'un destinataire ECTP reçoit un paquet CR, il invoque son démon récepteur RSVP associé;
- (7) à partir du message PATH qui reçoit en provenance de l'expéditeur, un récepteur RSVP élabore les messages RESV correspondants contenant les paramètres RSPEC. Une partie de l'information sur les paramètres RSPEC tels *R* (largeur de bande) et *S* (marge) peut être configurée par interaction entre le destinataire RSVP et le destinataire ECTP;
- (8) un destinataire RSVP transmet les messages RESV correspondant à l'expéditeur;
- (9) en cas d'indication d'erreur RESV RSVP, le destinataire RSVP communique le code d'état correspondant au destinataire ECTP.

Lorsque l'ECTP reçoit un code d'état de son RSVP associé, il peut communiquer cette information à son application. L'utilisation détaillée des codes d'état dépend des implémentations.

La Figure A.3 illustre les flux de messages ECTP et RSVP entre l'expéditeur et les destinataires. Après avoir transmis un paquet HB, l'expéditeur ECTP peut émettre un message initial PATH RSVP. Les messages correspondants RESV RSVP lui seront transmis par les destinataires ECTP. L'émission de messages PATH RSVP sera déclenchée et répétée avec des paquets HB ECTP périodiques.

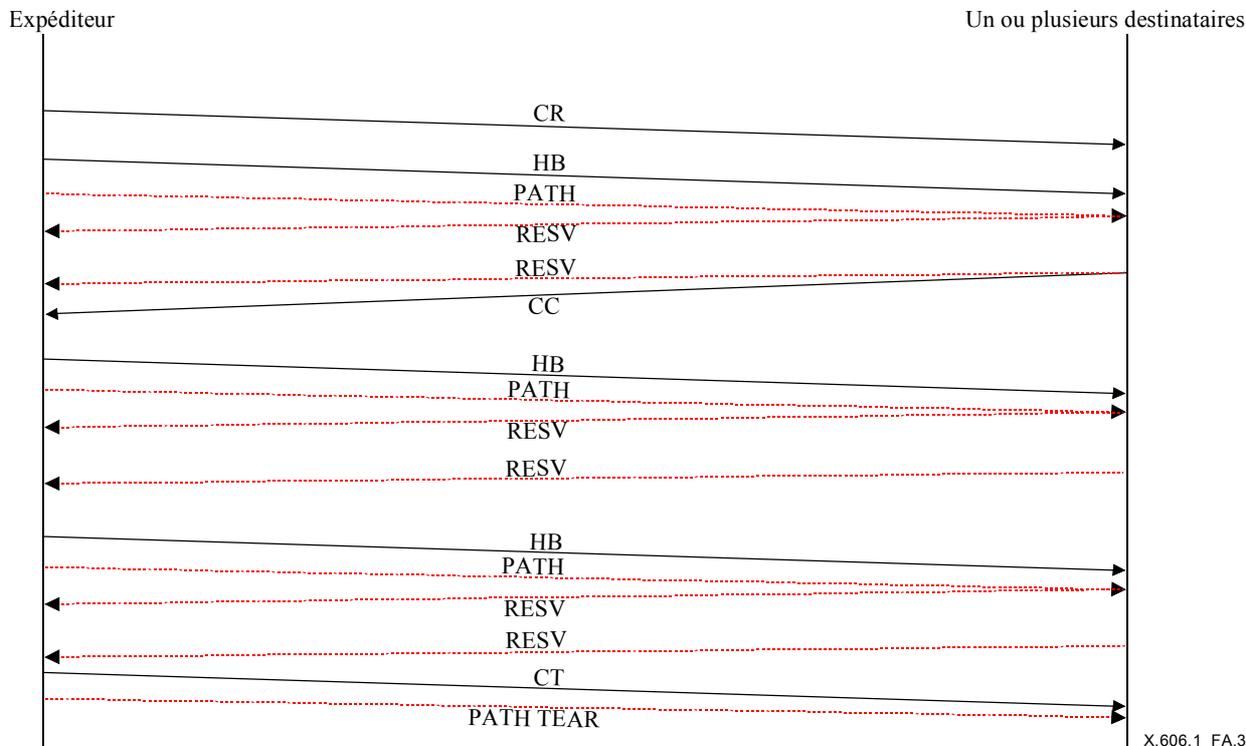


Figure A.3 – Modèle d'interfonctionnement entre protocoles ECTP et RSVP

Annexe B

Interfaces de programmation d'application

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

La présente annexe spécifie les interfaces de programmation d'application (API) pour le protocole ECTP. L'interface API ECTP décrite dans la présente Recommandation | Norme internationale peut être utilisée par les applications qui recourent aux capacités de transport du protocole ECTP partie 1 (Rec. UIT-T X.606 | ISO/CEI 14476-1) et du protocole ECTP partie 2 (Rec. UIT-T X.606.1 | ISO/CEI 14476-2).

Cette interface API a été conçue sur la base des fonctions API connecteur Berkeley socket. Toutefois, pour différencier l'interface API ECTP des fonctions connecteur Berkeley socket existantes, les fonctions de l'interface API ECTP sont nommées avec un préfixe "m" (par exemple, msocket).

B.1 Aperçu général

B.1.1 Fonctions de l'interface API

Le Tableau B.1 contient un récapitulatif des fonctions API utilisées pour le protocole ECTP.

Tableau B.1 – Fonctions de l'interface API ECTP

Nom de la fonction	Description
msocket ()	Crée un nouveau connecteur de multidiffusion dans le domaine des communications ECTP.
Mbind ()	Associe au connecteur un ensemble d'adresses/de ports locaux et de groupe.
Mconnect ()	L'expéditeur déclenche une création d'une connexion vers une adresse étrangère spécifiée. Le participant tardif déclenche un processus de ralliement.
maccept ()	Les destinataires possibles se rallient à la connexion ECTP en acceptant le signal de création d'une connexion depuis l'expéditeur.
Msend ()	Envoie les données d'application à un groupe de destinations.
Mrecv ()	Remet les données reçues à l'application. Remet certains messages d'indication destinés à la commande à l'application pendant la phase de transfert de données.
mclose ()	Met fin à la connexion et libère le connecteur.
mgetsockopt ()	Extrait du noyau les options de connecteur et de protocole.
msetsockopt ()	Fixe les options de connecteur et de protocole en direction du noyau.

B.1.2 Utilisation des fonctions API ECTP

La Figure B.1 illustre un exemple d'utilisation des fonctions API ECTP. Les séquences des fonctions API invoquées sont représentées en termes d'expéditeur, de destinataire et de destinataire tardif.

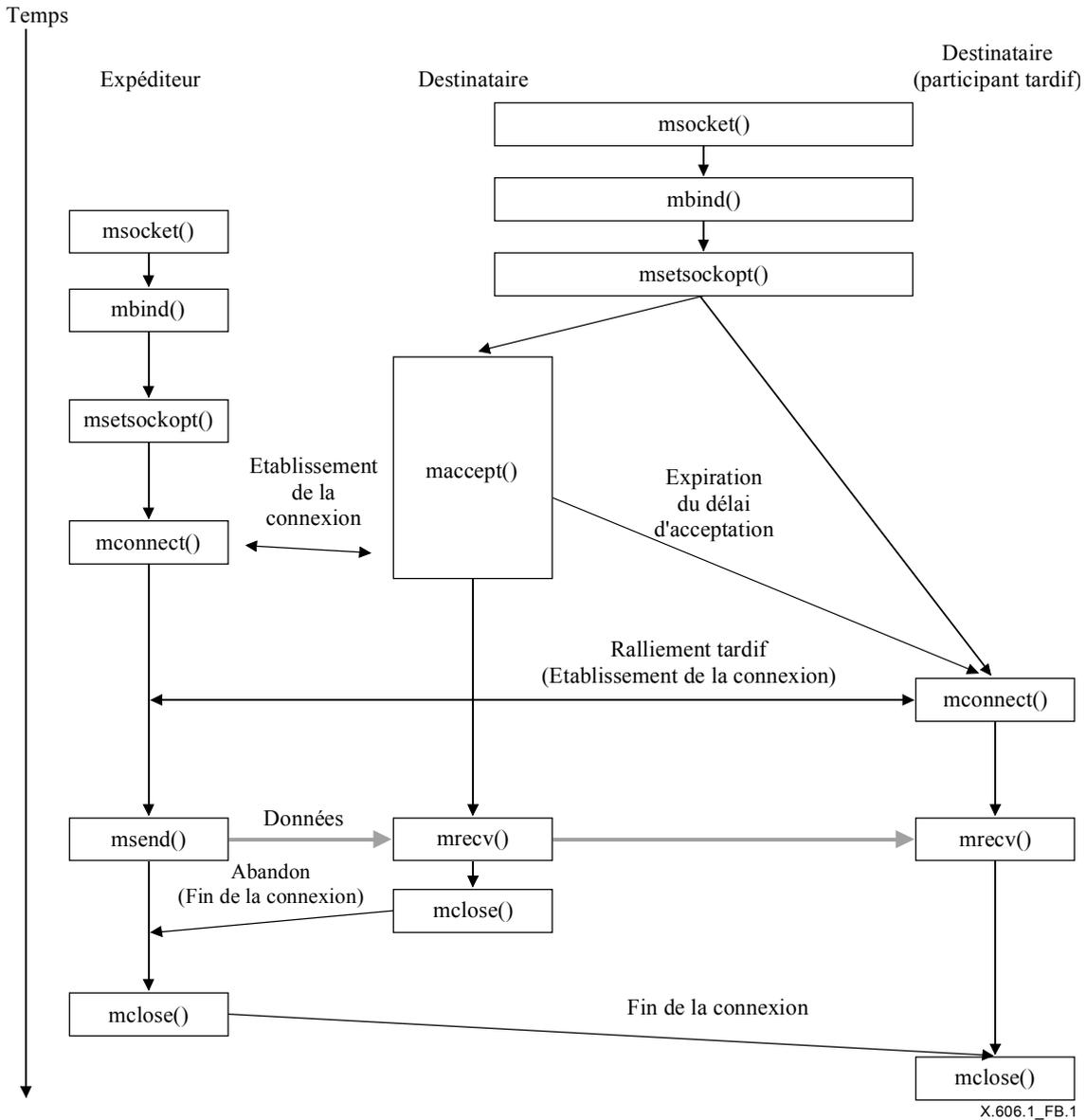


Figure B.1 – Utilisation de l'interface API ECTP

Comme le montre la figure, un destinataire procède à des actions différentes en fonction du type de noeud dont il relève: un participant précoce ou un participant tardif. Un participant précoce passe au mode `maccept` après le mode `msetsockopt`, tandis qu'un participant tardif invoque directement une fonction `mconnect`. Un participant précoce peut passer au mode `mconnect` après expiration du délai d'acceptation spécifiée, s'il ne peut recevoir de signal d'indication de création de connexion.

B.2 Fonctions API ECTP

B.2.1 Fonction `msocket()`

Pour utiliser le protocole ECTP, une application DOIT tout d'abord invoquer la fonction `msocket`, qui spécifie le type de protocole de communication souhaité tel l'ECTP utilisant l'IPv4, l'ECTP utilisant l'IPv6, l'ECTP utilisant les protocoles OSI, etc.

```
int msocket(int family, int type, int protocol);
```

Description des paramètres:

- *family*: spécifie la famille de protocole et est l'une des constantes présentées dans le Tableau B.2;
- *type*: spécifie le type de connecteur et l'une des constantes représentées dans le Tableau B.3;
- *protocol*: mis à 0.

Tableau B.2 – Constantes protocolaires *family* utilisées avec la fonction `msocket`

family	Description
AF_INET	Protocoles IPv4
AF_INET6	Protocoles IPv6
AF_ISO	Protocoles du domaine OSI

Tableau B.3 – Paramètre *type* de connecteur utilisé avec la fonction `msocket`

type	Description
SOCK_ECTP1	Connecteur pour l'ECTP partie 1
SOCK_ECTP2	Connecteur pour l'ECTP partie 2

Le Tableau B.4 représente les combinaisons valides ainsi que le protocole réel qui est sélectionné par la paire.

Tableau B.4 – Combinaisons des paramètres *family* and *type* pour la fonction `msocket`

type	AF_INET	AF_INET6	AF_ISO
SOCK_ECTP1	ECTP1	ECTP1	ECTP1
SOCK_ECTP2	ECTP2	ECTP2	ECTP2

L'appel `msocket` renvoie un descripteur non négatif en cas de succès ou à -1 dans les cas des causes d'erreur suivantes dont la liste est donnée dans le Tableau B.5.

Tableau B.5 – Codes d'erreur associés à un appel `msocket`

Code d'erreur	Description
EPROTONOSUPPORT	Le type de protocole ou le protocole spécifié n'est pas pris en charge dans le domaine considéré.
EMFILE	Le tableau de descripteurs par processus est plein.
ENFILE	Le tableau fichier système est plein.
EACCES	Autorisation de créer un connecteur du type et/ou protocole spécifié est refusée.
ENOBUFS	Espace tampon insuffisant.

B.2.2 Fonction `mbind()`

La fonction `mbind` assigne à un connecteur un ensemble d'adresses locales, de groupe ou de commande ainsi que le rôle du nœud dans la session. Dans le cas de protocoles Internet, l'adresse protocolaire est la combinaison d'une adresse IPv4 à 32 bits ou d'une adresse IPv6 à 128 bits, ainsi qu'un numéro de port de 16 bits.

```
int mbind(int msockfd, const struct sockaddr *laddr, socklen_t laddrlen, const struct
sockaddr *gaddr, socklen_t gaddrlen, struct sockaddr *caddr, socklen_t caddrlen, int
role );
```

Description des paramètres:

- *msockfd*: descripteur de connecteur qui a été renvoyé par la fonction *msocket*;
- *laddr*: pointeur sur une adresse spécifique au protocole pour associer une adresse locale au connecteur ci-dessus;
- *laddrlen*: taille de la structure d'adresse ci-dessus;
- *gaddr*: pointeur sur une adresse propre au protocole destiné à associer une adresse de groupe cible au connecteur;
- *gaddrlen*: taille de la structure d'adresse de groupe;
- *caddr*: pointeur vers une adresse spécifique au protocole destiné à associer une adresse de commande au connecteur;
- *caddrlen*: taille de la structure d'adresse de commande;
- *role*: spécifie le rôle de l'initiateur appelant tel TO, LO ou LE.

Tableau B.6 – Paramètre *role* de l'utilisateur du connecteur pour la fonction *msocket*

role	Description
TO	Créateur de la connexion et expéditeur en qualité de propriétaire dans les communications ECTP.
LO	Destinataire prenant la responsabilité des transmissions dans une hiérarchie arborescente.
LE	Destinataire qui n'est pas désigné par LO.

Une application peut *mbind*, c'est-à-dire associer une adresse IP spécifique et une adresse de groupe de réseau à son connecteur associé. Les adresses de la source et du groupe doivent appartenir à une interface sur le serveur.

L'appel *mbind* renvoie un zéro en cas de succès ou -1 dans le cas des causes d'erreur suivantes dont la liste est donnée dans le Tableau B.7.

Tableau B.7 – Codes d'erreur associés à la fonction *mbind*

Code d'erreur	Description
EAGAIN	Les ressources du noyau nécessaires pour répondre à la demande sont provisoirement indisponibles.
EBADF	<i>msockfd</i> n'est pas un descripteur valide.
ENOTSOCK	<i>msockfd</i> n'est pas un connecteur.
EADDRNOTAVAIL	L'adresse spécifiée n'est pas disponible depuis la machine locale.
EADDRINUSE	L'adresse spécifiée est déjà utilisée.
EACCES	L'adresse demandée est protégée et l'utilisateur actuel n'a pas d'autorisation valide pour y accéder.
EFAULT	Le paramètre d'adresse n'est pas dans une partie valide de l'espace adresse de l'utilisateur.
EROLE	Le rôle demandé n'est pas valide.
NOTE – Le code d'erreur ombré est nouvellement défini pour le protocole ECTP.	

B.2.3 Fonction `maccept()`

Seul le membre passif d'une session tel le LE ou le LO peut invoquer cette fonction. Il peut en attendre l'initiation par l'expéditeur pendant un délai spécifique via la fonction `msetsockopt` et indiquer si la connexion à multidiffusion a été établie ou non.

```
int maccept(int msockfd, struct sockaddr *raddr, socklen_t *raddrlen);
```

Description des paramètres:

- *msockfd*: descripteur de connecteur qui a été renvoyé par la fonction `msocket`;
- *raddr*: renvoie l'adresse protocolaire de l'initiateur de la connexion distante (l'expéditeur ou le propriétaire TO);
- *raddrlen*: pointeur sur la taille de la structure d'adresse du connecteur pointé par *raddr*.

Si l'application de la fonction `maccept` donne un résultat positif, elle renvoie la même valeur comme premier argument, *msockfd*. Après cela, nous appelons cette valeur renvoyée descripteur *connected socket*.

L'appel `maccept` renvoie un descripteur non négatif en cas de succès ou la valeur `-1` dans le cas des causes d'erreur suivantes dont la liste est donnée dans le Tableau B.8.

Tableau B.8 – Codes d'erreur associés à la fonction `maccept`

Code d'erreur	Description
EBADF	Le descripteur n'est pas valide.
EINTR	L'opération <code>maccept</code> a été interrompue.
EMFILE	Le tableau de descripteur par processus est plein.
ENFILE	Le tableau fichier système est plein.
ENOTSOCK	Les références du descripteur sont un fichier et non pas un connecteur.
EFAULT	Le paramètre <i>addr</i> ne se trouve pas dans une partie inscriptible de l'espace d'adresse d'utilisateur.
EWOULDBLOCK	Le connecteur est marqué comme étant non bloquant et aucune connexion n'est présente pour pouvoir être acceptée.
ECONNABORTED	Une connexion est arrivée, mais elle a été fermée pendant qu'elle se trouvait dans la file d'attente d'écoute
ECRTIMEOUT	Indique que le délai d'attente de la demande CR a expiré.
NOTE – Le code d'erreur ombré est nouvellement défini pour le protocole ECTP.	

B.2.4 Fonction `mconnect()`

La fonction `mconnect` est utilisée par le propriétaire TO ou par le LE tardif pour établir une connexion.

```
int mconnect(int msockfd, const struct sockaddr *daddr, socklen_t daddrlen);
```

Description des paramètres:

- *msockfd*: descripteur de connecteur qui a été renvoyé par la fonction `msocket`;
- *daddr*: pointeur vers une adresse de destination propre au protocole. Cette adresse peut être un groupe d'adresses (dans le cas où l'initiateur appelant est un propriétaire TO), ou une adresse d'expéditeur (dans le cas où l'initiateur appelant est le LO ou le LE);
- *daddrlen*: taille de *daddr*.

La fonction `mconnect` renvoie un zéro en cas de succès ou `-1` dans les cas anormaux dont la liste est donnée dans le Tableau B.9.

Tableau B.9 – Codes d'erreur associés à la fonction `mconnect`

Code d'erreur	Description
EBADF	<i>msockfd</i> n'est pas un descripteur valide.
ENOTSOCK	<i>msockfd</i> est un descripteur pour un fichier et non pas un connecteur.
EADDRNOTAVAIL	L'adresse spécifiée n'est pas disponible sur cette machine.
EAFNOSUPPORT	Les adresses de la famille d'adresses spécifiées ne peuvent être utilisées avec ce connecteur.
EISCONN	Le connecteur est déjà connecté.
ECONNREFUSED	Une tentative de connexion a été vigoureusement rejetée.
ENETUNREACH	Le réseau n'est pas atteignable à partir de ce serveur.
EADDRINUSE	L'adresse est déjà utilisée.
EFAULT	Le paramètre nom spécifie une zone en dehors de l'espace adresse du processus.
EALREADY	Le connecteur n'est pas bloquant et il n'a pas été mis fin à une précédente tentative de connexion.
EDENIED	Indique que le TO a décliné la demande de ralliement du LE ou du LO.
ETIMEDOUT	La temporisation associée à l'établissement d'une connexion est expirée. Indique qu'il n'y a pas de réponse provenant du TO.
NOTE – Les codes d'erreur ombrés sont nouvellement définis pour le protocole ECTP.	

B.2.5 Fonction `msend()`

Cette fonction `msend` écrit dans un connecteur connecté les données provenant d'un tampon.

```
ssize_t msend (int msockfd, const void *buf, size_t buflen, int *flags);
```

Description des paramètres:

- *msockfd*: descripteur de connecteur qui a été renvoyé par la fonction `msocket`;
- *buf*: pointeur sur un tampon à partir duquel l'écriture aura lieu;
- *buflen*: taille de *buf*;
- *flags*: non encore défini.

La fonction `msend` renvoie le nombre d'octets écrits en cas de succès ou `-1` dans les cas d'erreurs dont la liste est donnée dans le Tableau B.10.

Tableau B.10 – Codes d'erreur associés à la fonction `msend`

Code d'erreur	Description
EBADF	Un descripteur non valide a été spécifié.
EACCES	L'adresse de destination est une adresse de multidiffusion et le connecteur n'a pas encore été réglé sur <code>SO_BROADCAST</code> .
ENOTSOCK	L'argument <i>msockfd</i> n'est pas un connecteur.
EFAULT	Une adresse d'espace utilisateur non valide a été spécifiée pour un paramètre.
EMSGSIZE	Le connecteur exige que le message soit envoyé de manière atomique, et la taille du message à envoyer n'a pas permis de le faire.
EAGAIN	Le connecteur est marqué comme étant non bloquant et l'opération demandée produirait un blocage.
ENOBUFS	Le système n'a pas été en mesure d'attribuer un tampon interne. Cette opération pourra réussir lorsque des tampons seront disponibles.
ENOBUFS	La file d'attente de sortie d'une interface de réseau est pleine. Cela indique généralement que l'interface a cessé d'émettre, mais peut être dû à un encombrement transitoire.
EPARTITIONED	Indique que la session a fait l'objet d'une partition.
NOTE – Le code d'erreur ombré est nouvellement défini pour le protocole ECTP.	

B.2.6 Fonction mrecv()

La fonction `mrecv` est utilisée pour recevoir des données multidiffusées et des signaux d'indication à des fins de commande.

```
ssize_t mrecv (int msockfd, void *buf, size_t buflen, int *flags, struct sockaddr
*fromaddr, socklen_t *fromaddrlen);
```

Description des paramètres:

- *msockfd*: descripteur de connecteur qui a été renvoyé par la fonction `msocket`;
- *buf*: pointeur sur un tampon dans lequel une lecture doit avoir lieu;
- *buflen*: taille de *buf*;
- *flags*: non encore défini;
- *fromaddr*: pointeur vers une adresse propre au protocole pour spécifier l'expéditeur;
- *fromaddrlen*: taille de *fromaddr*.

Lorsqu'une application reçoit des données en provenance du tampon, elle peut identifier l'expéditeur correspondant en utilisant le paramètre *fromaddr*.

La fonction `mrecv` renvoie le nombre d'octets reçus en cas de succès. Dans les autres cas, elle renvoie la valeur `-1`, lorsqu'une erreur s'est produite ou lorsqu'un message de commande a été remis à une application. La liste des codes d'erreur est donnée dans le Tableau B.11.

Tableau B.11 – Codes d'erreur associés à la fonction `mrecv`

Code d'erreur	Description
EBADF	L'argument <i>msockfd</i> est un descripteur non valide.
ENOTCONN	Le connecteur est associé avec un protocole orienté connexion et n'a pas été connecté (voir les fonctions <code>mconnect</code> et <code>maccept</code>).
ENOTSOCK	L'argument <i>msockfd</i> ne renvoie pas à un connecteur.
EAGAIN	Le connecteur est marqué comme étant non bloquant et l'opération de réception provoquera un blocage, ou un temporisateur de réception a été armé et la temporisation a expiré avant que les données aient été reçues.
EINTR	La réception a été interrompue par la remise d'un signal avant que des données aient été disponibles.
EFAULT	Le ou les pointeurs du tampon de réception pointent en dehors de l'espace adresse du processus.
ETOTERM	Le TO a mis fin à la session.
ETOEXPEL	Le TO a éjecté le LE ou le LO.
EPARTITIONED	Indique que la session a fait l'objet d'une partition.
NOTE – Les codes d'erreur ombrés sont nouvellement définis pour le protocole ECTP.	

B.2.7 Fonction mclose()

La fonction `mclose` est utilisée pour quitter ou mettre fin à une connexion ECTP par fermeture du connecteur. L'action par défaut de `mclose` avec un connecteur ECTP consiste à marquer le connecteur comme étant fermé et revenir au processus immédiatement. Le descripteur de connecteur n'est plus utilisable par le processus.

```
int mclose (int msockfd);
```

Description du paramètre:

- *msockfd*: descripteur de connecteur qui a été renvoyé par la fonction `msocket`.

La fonction `mclose` renvoie un zéro en cas de succès ou -1 en cas d'erreur, comme indiqué dans le Tableau B.12.

Tableau B.12 – Codes d'erreur associés à la fonction `mclose`

Code d'erreur	Description
EBADF	<i>msockfd</i> n'est pas un descripteur actif.
EINTR	Une interruption a été reçue.

B.2.8 Fonctions `mgetsockopt()` et `msetsockopt()`

La fonction `mgetsockopt` est utilisée pour déterminer les options et les caractéristiques de la connexion actuelle qui affecte un connecteur.

```
int mgetsockopt(int msockfd, int level, int optname, void *optval, socklen_t *optlen);
```

Description des paramètres:

- *msockfd*: descripteur de connecteur qui a été retourné par la fonction `msocket`;
- *level*: spécifie le code dans le système qui sert à interpréter l'option: le code général de connecteur ou certains codes propres au protocole (par exemple, IPv4, IPv6 ou ECTP);
- *optname*: spécifie le type d'option, chaque niveau peut définir plusieurs noms d'option;
- *optval*: pointeur sur une variable dans laquelle la valeur courante de l'option est stockée par `mgetsockopt`;
- *optlen*: spécifie la longueur de *optval* comme le résultat de valeur pour `mgetsockopt`.

La fonction `msetsockopt` est utilisée pour déterminer les options qui affectent un connecteur.

```
int msetsockopt(int msockfd, int level, int optname, const void *optval, socklen_t *optlen);
```

Description des paramètres:

- *msockfd*: descripteur de connecteur qui a été retourné par la fonction `msocket`;
- *level*: spécifie le code dans le système permettant d'interpréter l'option: le code général de connecteur ou certains codes propres au protocole (par exemple, IPv4, IPv6 ou ECTP);
- *optname*: spécifie le type d'option. Chaque niveau peut définir plusieurs noms d'option;
- *optval*: pointeur sur une variable dans laquelle la valeur courante de l'option est stockée par la fonction `msetsockopt`;
- *optlen*: spécifie comme valeur la longueur de *optval* pour la fonction `msetsockopt`.

Le Tableau B.13 est un récapitulatif des options qui peuvent être demandées par les fonctions `mgetsockopt` et `msetsockopt`.

Tableau B.13 – Options connecteur pour `mgetsockopt` et `msetsockopt` avec le protocole ECTP

Niveau	optname	Mget	mset	Description	Indicateur	Type de données
IPPROTO_ECTP	ECTP_QOS	•	•	Extrait et donne des valeurs aux paramètres de QS		QoS
	ECTP_OPPAR		•	Extrait et indique l'adresse du parent de commande		OP_par
	ECTP_OPVAR1	•	•	Extrait et fixe les valeurs d'opération pour l'ECTP partie I		OP_var1
	ECTP_OPVAR2	•	•	Extrait et fixe les valeurs d'opération pour l'ECTP partie II		OP_var2
	ECTP_OPLJ	•	•	Active le participant tardif	•	int
	ECTP_OPWCR	•	•	Fixe le délai d'attente du message CR		u_long

Les fonctions `mgetsockopt` et `msetsockopt` renvoient un zéro en cas de succès ou `-1` en cas d'erreur comme indiqué dans le Tableau B.14.

Tableau B.14 – Codes d'erreur associés aux fonctions `mgetsockopt` et `msetsockopt`

Code d'erreur	Description
EBADF	L'argument <code>msockfd</code> n'est pas un descripteur valide.
ENOTSOCK	L'argument <code>msockfd</code> est un fichier et non pas un connecteur.
ENOPROTOOPT	L'option est inconnue au niveau indiqué.
EFAULT	L'adresse pointée par <code>optval</code> n'est pas une partie valide de l'espace adresse du processus. Pour <code>mgetsockopt</code> , cette erreur peut être également retournée si <code>optlen</code> n'est pas une partie valide de l'espace adresse du processus.

Les structures de données suivantes sont illustrées sous forme d'informations concernant l'utilisation des fonctions `mgetsockopt` ou `msetsockopt`.

```
typedef struct _op_var1 {
    int agn; /* ACK Generation Number (Numéro de génération ACK) */
    int mcn; /* Maximum Children Number (Numéro maximal des enfants) */
    int mrn; /* Maximum Retransmission Number (Numéro maximal de retransmission) */
    int mtl; /* Maximum Tree Level (Niveau maximal de l'arborescence) */
    int nft; /* Node Failure Threshold (Seuil de défaillance du nœud) */
    int arn; /* Active Receiver Number (Numéro du destinataire actif) */
    int ccn; /* Current Children Number (Numéro des enfants actuels) */
    int crn; /* Current Receiver Number (Numéro du destinataire actuel) */
    int ctl; /* Current Tree Level (Niveau de l'arborescence actuel) */
    int ctr; /* Current Transmission Rate (Débit de transmission actuel) */
    int cct; /* Connection Creation Time (Instant de création de la connexion) */
}OP_var1;
```

```
typedef struct _op_var2 {
    int mintr; /* Minimum Transmission Rate (Débit minimal de transmission) */
    int ctr; /* Current Transmission Rate (Débit actuel de transmission) */
    int maxtr; /* Maximum Transmission Rate (Débit maximal de transmission) */
}OP_var2;
```

```

typedef struct _op_par{
    u_long  ctladdr; /* Local Group Control Multicast Address (Adresse de multidiffusion du groupe
    local de commande) */
    u_short ctlport; /* Local Group Control Port (Port du groupe local de commande) */
}OP_par;

```

```

typedef struct _QoS {
    u_short threshold; /* threshold rate (Débit seuil) */
    u_long  th_CHQ_TQA:2,th_CHQ_N:1,th_CHQ:29; /* CHQ throughput (Débit CHQ) */
    u_long  th_OT_TQA:2,th_OT_N:1,th_OT:29; /* OT throughput (Débit OT) */
    u_long  th_LQA_TQA:2,th_LQA_N:1,th_LQA:29; /* LQA throughput (Débit LQA) */
    u_long  td_OT_TQA:2,td_OT_N:1,td_OT:29; /* OT transit delay (Temps de
    transit OT) */
    u_long  td_LQA_TQA:2,td_LQA_N:1,td_LA:29; /* LQA transit delay (Temps de
    transit LQA) */
    u_long  tdj_OT_TQA:2,tdj_OT_N:1,tdj_OT:29; /* OT transit delay jitter (Gigue de temps
    de transit OT) */
    u_long  tdj_LQA_TQA:2,tdj_LQA_N:1,tdj_LQA:29; /* LQA transit delay jitter (Gigue de
    temps de transit LQA) */
    u_long  cr_TSDU_TQA:2,cr_TSDU_N:1,cr_TSDU:29; /* corrupted data rate (Taux de données
    erroné) */
    u_long  l_TSDU_TQA:2,l_TSDU_N:1,l_TSDU:29; /* lost data rate (Taux de données
    perdu) */
}QoS;

```

B.3 Exemple d'un fichier d'en-tête msocket.h

Ce qui suit décrit le fichier d'en-tête "msocket.h" qui est utilisé dans une réalisation du protocole ECTP pour information.

```

#ifndef __MSOCKET_H_
#define __MSOCKET_H_
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/un.h>
#include "ectpcb.h"
#include "var.h"
#define SIZE_MSETSOCKOPT 524
struct _common{
    u_char command;
    u_short len;
    u_long error;
};
enum{ MCONNECT = 101,
    MBIND,
    MSEND,
    MRECV,
    MACCEPT,
    MLISTEN,
    MCLOSE,
    MSETSOCKOPT,
    MGETSOCKOPT,
    DETACH};

```

```

/* valeur msocket retournée */
#define ECONNECT_OK      0x00
#define ECONNECT_ROLE  0x01
#define ESEND_OK        0x0000
#define ESEND_RESEND   0x1004
#define ESEND_NOSEND   0x1001
#define ESEND_NOONE    0x1002
#define EBIND_OK        0x00
#define EBIND_ADDR      0x01
#define EBIND_ROLE     0x02

#define ECLOSE_OK       0x00
#define ECLOSE_TO      0x01 // Lorsque TO est invoqué
#define ECTP_QOS       0x4001
#define ECTP_OPVAR1   0x4002
#define ECTP_OPVAR2   0x4003
#define ECTP_OPLJ     0x4004
#define ECTP_OPWCR     0x4005
#define ECTP_OPTREE    0x4006
#define ECTP_OPTIME    0x4007
#define ECTP_OPFLOW    0x4008
#define ECTP_OPPAR    0x4009

/* types de message msocket */
#define FIXED_SIZE     24 // taille de fixed_header
struct _mbind_req{
    struct _common c;
    int role;
    struct sockaddr_in ctrl;
    struct sockaddr_in local;
    struct sockaddr_in grp;
};
struct _mconnect_req{
    struct _common c;
};
struct _maccept_req{
    struct _common c;
    struct sockaddr_in to;
};

struct _msend_rep{
    struct _common c;
};

```

```

struct _mrecv_req{
    struct _common c;
};
struct _msend_req{
    struct _common c;
    u_char rsvd[FIXED_SIZE - sizeof(struct _common)];
    u_char data[MAXDATA];
};
struct _mrecv_rep{
    struct _common c;
    u_char rsvd[FIXED_SIZE - sizeof(struct _common)];
    u_char data[MAXDATA];
};
struct _mclose_req{
    struct _common c;
};
struct _mopt_op{
    int agn;
    int arn;
    int ccn;
    int ctl;
    int mcn;
    int mrn;
    int mtl;
    int nft;
    int lrft;
};
struct _mopt_time{
    long agt; // ACK Generation time (Instant de génération de l'ACK)
    long cct; // Connection Creation time (Instant de création de la connexion)
    long hgt; // HB Generation time (Instant de génération HB)
    long iat; // Inactivity time (Temps d'inactivité)
    long ndt; // ND Generation time (Instant de génération ND)
    long rbt; // Retransmission Backoff time (Instant de réduction de la retransmission)
    long rxt; // Retransmission time (Durée de la retransmission)
    long tct; // Tree Creation time (Instant de création de l'arborescence)
};
struct _mopt_flow{
    int cit;
    int cmn;
    int crn;
    int mintr;
    int ctr;
    int maxtr;
    int iri;
    int cri;
    int crd;
};
struct _mopt_tree{

```

```

int tree_ct; // connection type (Type de connexion)
int tree_conf; // Tree Level Configuration (Configuration du niveau de l'arborescence)
int tree_mtl; // Max Tree Level (only valid if tree_conf ==2) (Niveau maximal de l'arborescence
              // (valide seulement si tree_conf ==2))
int tree_mcn; // Max Children Number (Nombre maximal d'enfants)
int bitmapsize; // ack bitmap size (Taille bitmap d'ACQ)
};

struct _mgetsockopt_req{
    struct _common c;
    int    optname;
    int    optlen;
    u_char optval[MAXDATA/2];
};

struct _msetsockopt_req{
    struct _common c;
    int    optname;
    int    optlen;
    u_char optval[MAXDATA/2];
};

char cbreq[128];
char dbreq[4096];
#endif

```

Bibliographie

Les Normes IETF RFC sont utiles pour comprendre ou implémenter la présente Spécification:

- IETF RFC 768, *User Datagram Protocol, Internet Standard*, août 1980.
- IETF RFC 791, *Internet Protocol, DARPA Internet Program, Protocol specification, Internet Standard*, septembre 1981.
- IETF RFC 793, *Transmission Control Protocol, DARPA Internet Program, Protocol specification, Internet Standard*, septembre 1981.
- IETF RFC 1112, *Host Extensions for IP Multicasting, Internet Standard*, août 1989.
- IETF RFC 1119, *Network Time Protocol, Internet Standard*, mai 1990.
- IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels, Best Current Practice*, mars 1997.
- IETF RFC 2205, *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, Proposed Standard*, septembre 1997.
- IETF RFC 2210, *The Use of RSVP with IETF Integrated Services, Proposed Standard*, septembre 1997.
- IETF RFC 2236, *Internet Group Management Protocol, Version 2, Proposed Standard*, novembre 1997.
- IETF RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification, Draft Standard*, décembre 1998.
- IETF RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, Proposed Standard*, décembre 1998.
- IETF RFC 2597, *Assured Forwarding PHB Group, Proposed Standard*, juin 1999.
- IETF RFC 2598, *An Expedited Forwarding PHB, Proposed Standard*, juin 1999.
- IETF RFC 2750, *RSVP Extensions for Policy Control, Proposed Standard*, janvier 2000.
- IETF RFC 2836, *Per Hop Behavior Identification Codes, Proposed Standard*, mai 2000.

SÉRIES DES RECOMMANDATIONS UIT-T

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