

Union internationale des télécommunications

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**X.608**

(02/2007)

SÉRIE X: RÉSEAUX DE DONNÉES, COMMUNICATION  
ENTRE SYSTÈMES OUVERTS ET SÉCURITÉ

Réseautage OSI et aspects systèmes – Réseautage

---

**Technologies de l'information – Protocole  
de transport de communications amélioré:  
spécification du transport multidiffusé n-plex**

Recommandation UIT-T X.608



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

<b>RÉSEAUX PUBLICS DE DONNÉES</b>	
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
<b>INTERCONNEXION DES SYSTÈMES OUVERTS</b>	
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
Formulaire 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
<b>INTERFONCTIONNEMENT DES RÉSEAUX</b>	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.379
<b>SYSTÈMES DE MESSAGERIE</b>	<b>X.400–X.499</b>
<b>ANNUAIRE</b>	<b>X.500–X.599</b>
<b>RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES</b>	
<b>Réseautage</b>	<b>X.600–X.629</b>
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
<b>GESTION OSI</b>	
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
<b>SÉCURITÉ</b>	<b>X.800–X.849</b>
<b>APPLICATIONS OSI</b>	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.889
Applications génériques de l'ASN.1	X.890–X.899
<b>TRAITEMENT RÉPARTI OUVERT</b>	<b>X.900–X.999</b>
<b>SÉCURITÉ DES TÉLÉCOMMUNICATIONS</b>	<b>X.1000–</b>

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

**Technologies de l'information – Protocole de transport de communications amélioré:  
spécification du transport multidiffusé n-plex**

**Résumé**

La présente Recommandation | Norme internationale décrit un protocole de transport multidiffusé n-plex sur Internet lorsque la multidiffusion IP est prise en charge. Elle définit les mécanismes de gestion de session et de protection contre les erreurs. Pour la gestion de session, un participant est désigné pour gérer la création et la fin d'une connexion, l'arrivée et la sortie d'un participant et les jetons qui permettent aux différents participants d'envoyer des données. En ce qui concerne la protection contre les erreurs, la Recommandation définit les mécanismes de reprise après perte fondés sur une arborescence, la construction de l'arborescence de gestion au moyen d'une arborescence logique à deux couches et l'adaptation de l'arborescence logique à l'état de transmission des paquets. La présente Spécification décrit les détails du protocole, notamment le format des paquets, les procédures et les valeurs des paramètres. Le protocole peut être utilisé pour les applications qui nécessitent un service de transmission de données "de beaucoup à beaucoup" (*many-to-many*).>

**Source**

La Recommandation UIT-T X.608 a été approuvée le 13 février 2007 par la Commission d'études 17 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8. Un texte identique est publié comme Norme Internationale ISO/CEI 14476-5.

## 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.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

## 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 des renseignements les plus récents, il est vivement recommandé aux développeurs de consulter la base de données des brevets du TSB sous <http://www.itu.int/ITU-T/ipr/>.

© UIT 2008

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 ..... 1
2.1	Références normatives ..... 1
2.2	Références informatives ..... 1
3	Définitions ..... 1
4	Abréviations ..... 2
5	Conventions ..... 3
6	Aperçu général..... 3
7	Considérations ..... 6
7.1	Participants ..... 6
7.2	Voie de transmission de données et adressage ..... 6
7.3	Voie et arborescence de gestion ..... 7
7.4	Jetons ..... 9
7.5	Adaptation des arborescences logiques ..... 9
8	Paquets..... 11
8.1	En-tête de base ..... 11
8.2	Éléments d'extension ..... 14
8.3	Format de paquet ..... 18
9	Procédures ..... 39
9.1	Gestion de connexion..... 39
9.2	Gestion d'arborescence logique ..... 41
9.3	Transport de données multidiffusées ..... 46
9.4	Gestion des jetons ..... 49
9.5	Mesure du temps RTT..... 52
10	Paramètres du système..... 52
Annexe A	– Interfaces de programmation d'application ..... 53
A.1	Aperçu général ..... 53
A.2	Fonctions API du protocole ECTP-5 ..... 53
Annexe B	– Diagrammes de transition d'état..... 59
Annexe C	– Exemple de valeurs de paramètres système dans le protocole ECTP-5 ..... 61

## Introduction

Le protocole de transport de communications amélioré (ECTP, *enhanced communications transport protocol*) a pour objet de prendre en charge les connexions multidiffusion étroitement gérées dans des applications simplex, duplex et n-plex. La présente partie du protocole ECTP (partie 5: UIT-T X.608 | ISO/CEI 14476-5) définit les mécanismes protocolaires relatifs au transport de données multidiffusées n-plex.

Dans la connexion multidiffusée n-plex, les participants sont le propriétaire de la connexion TC et de nombreux utilisateurs du service de transport (TS). Le propriétaire de la connexion TC sera choisi parmi les utilisateurs du service TS avant le début de la connexion. Il est au centre des communications multidiffusées de groupe. Il est chargé de la gestion globale de la connexion et, pour ce faire, il gère la création et la fin de la connexion, le transport de données multidiffusées et les opérations de participation tardive ou de sortie.

Dans la connexion multidiffusée n-plex, les transmissions de données multidiffusées sont autorisées par les utilisateurs du service TS et par le propriétaire de la connexion TC. Chacun des utilisateurs du service TS est autorisé à envoyer des données multidiffusées au groupe uniquement s'il obtient un jeton du propriétaire de la connexion TC. Autrement dit, les transmissions de données multidiffusées des utilisateurs du service TS sont gérées par le propriétaire de la connexion TC.

La connexion multidiffusée n-plex définie dans la présente Recommandation | Norme internationale vise les applications multidiffusion "de beaucoup à beaucoup" dans lesquelles de nombreux participants (utilisateurs du service TS) souhaitent transmettre des données multidiffusées à tous les autres utilisateurs du service TS. Des exemples types de ces applications sont les "téléconférences", les "jeux en ligne à multiples utilisateurs", etc. Dans les applications de téléconférences, le propriétaire de la connexion TC peut agir en qualité de "serveur de conférence" et tous les autres participants (utilisateurs du service TS) peuvent envoyer aux autres participants des données multidiffusées, telles que voix, textes et images.

**NORME INTERNATIONALE  
RECOMMANDATION UIT-T**

**Technologies de l'information – Protocole de transport de communications amélioré:  
spécification du transport multidiffusé n-plex**

## **1 Domaine d'application**

La présente Recommandation | Norme internationale définit la connexion de transport multidiffusé n-plex dans laquelle tous les participants sont des utilisateurs du service TS et l'un d'entre eux le propriétaire de la connexion TC. La connexion de transport multidiffusé n-plex permet aux utilisateurs du service TS d'envoyer des données multidiffusées à tous les membres du groupe. Il convient de noter qu'un utilisateur du service TS est autorisé à envoyer des données multidiffusées au groupe uniquement s'il obtient un jeton du propriétaire de la connexion TC.

La présente Spécification décrit le protocole de prise en charge du transport multidiffusé n-plex, qui comprend la gestion de la connexion (établissement, fin, participation et sortie d'un utilisateur) et les mécanismes de gestion de la fiabilité concernant le transport des données multidiffusées.

## **2 Références**

### **2.1 Références normatives**

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valides 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 CEI 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.602 (2004) | ISO/CEI 16511:2005, *Technologies de l'information – Protocole de gestion de groupe*.
- 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*.
- Recommandation UIT-T X.606.1 (2003) | ISO/CEI 14476-2:2003, *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.607 (2007) | ISO/CEI 14476-3:2007, *Technologies de l'information – Protocole de transport de communications amélioré: spécification du transport multidiffusé duplex*.

### **2.2 Références informatives**

- Projet de Recommandation UIT-T X.607.1 | ISO/CEI 14476-4, *Technologies de l'information – Protocole de transport de communications amélioré: spécification de la gestion de la qualité de service pour le transport multidiffusé duplex*.

## **3 Définitions**

La présente Recommandation | Norme internationale est fondée sur les définitions ci-après, qui sont indiquées dans la Rec. UIT-T X.605 | ISO/CEI 13252 (service de transport de communications amélioré).

- a) connexion de transport: simplex, duplex et n-plex.

## ISO/CEI 14476-5:2008 (F)

La présente Recommandation | Norme internationale redéfinit les expressions suivantes utilisées dans la Rec. UIT-T X.605 | ISO/CEI 13252 (service de transport de communications amélioré).

- a) **propriétaire de la connexion TC (TCN, *TC-owner*)**: le propriétaire de la connexion TC (TCN) gère les opérations globales d'une connexion multidiffusée n-plex;
- b) **utilisateur du service TS (utilisateur du service de transport)**: les utilisateurs du service TS peuvent envoyer et recevoir les données multidiffusées dans la connexion multidiffusée n-plex;
- c) **utilisateur expéditeur du service TS (SU, *sending TS-user*)**: utilisateur du service TS qui obtient un jeton auprès du propriétaire de la connexion TC. Seul l'utilisateur expéditeur SU est autorisé à envoyer des données multidiffusées au groupe. Autrement dit, avant d'envoyer les données multidiffusées, chaque utilisateur du service TS doit demander un jeton au propriétaire de la connexion TC.

La présente Recommandation | Norme internationale redéfinit les termes ci-après utilisés dans la Rec. UIT-T X.606 | ISO/CEI 14476-1 (Protocole de transport de communications amélioré: partie 1) afin de prendre en compte la connexion multidiffusée n-plex.

- a) **groupe local**: ensemble de nœuds proches ayant une corrélation de couche de réseau en termes de perte de paquets et de retard;
- b) **propriétaire local (LO, *local owner*)**: le propriétaire local est un nœud représentatif d'un groupe local désigné de manière statique. Il est chargé de mettre à jour une arborescence intra-groupe du groupe et des arborescences de gestion pour tous les utilisateurs SU du groupe local. Chaque propriétaire local est aussi connecté aux autres propriétaires locaux le long d'arborescences inter-groupes. Périodiquement, il crée aussi un trafic de test pour l'adaptation des arborescences logiques;
- c) **voie de transmission des données multidiffusées**: le propriétaire de la connexion TC (TCN) ou l'utilisateur expéditeur SU peut envoyer des données multidiffusées à tous les autres membres du groupe à une adresse IP multidiffusion.

La présente Recommandation | Norme internationale donne une nouvelle définition des termes ci-après:

- a) **arborescence logique**: arborescence qui englobe tous les utilisateurs du service TS et dont découlent une ou plusieurs arborescences de gestion;
- b) **arborescence inter-groupes**: arborescence logique, par source, des propriétaires locaux;
- c) **arborescence intra-groupe**: arborescence logique partagée par chaque groupe local;
- d) **arborescence de gestion**: arborescence parcourue par les paquets de gestion en vue de la protection contre les erreurs;
- e) **jeton**: représente le droit d'un utilisateur du service TS de transmettre des données multidiffusées. L'utilisateur du service TS détenteur d'un jeton est un utilisateur expéditeur SU. Les jetons sont gérés par le propriétaire de la connexion TC.

## 4 Abréviations

Les acronymes suivants sont employés dans la présente Recommandation | Norme internationale pour les protocoles ECTP:

ECTP-1	Protocole ECTP, partie 1 (UIT-T X.606   ISO/CEI 14476-1)
ECTP-2	Protocole ECTP, partie 2 (UIT-T X.606.1   ISO/CEI 14476-2)
ECTP-3	Protocole ECTP, partie 3 (UIT-T X.607   ISO/CEI 14476-3)
ECTP-4	Protocole ECTP, partie 4 (UIT-T X.607.1   ISO/CEI 14476-4)
ECTP-5	Protocole ECTP, partie 5 (UIT-T X.608   ISO/CEI 14476-5)
ECTP-6	Protocole ECTP, partie 6 (UIT-T X.608.1   ISO/CEI 14476-6)

Les acronymes suivants sont employés dans la présente Recommandation | Norme internationale pour les paquets ECTP-5:

ACK	accusé de réception ( <i>acknowledgment</i> )
CC	confirmation de création de connexion ( <i>connection creation confirm</i> )
CCC	confirmation de changement d'arborescence de gestion ( <i>control tree change confirm</i> )
CCR	demande de changement d'arborescence de gestion ( <i>control tree change request</i> )
CR	demande de création de connexion ( <i>connection creation request</i> )
CT	demande de fin de connexion ( <i>connection termination request</i> )

DT	données ( <i>data</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 d'un utilisateur ( <i>user leave request</i> )
NACK	accusé de réception négatif ( <i>negative acknowledgement</i> )
PB	sonde ( <i>probe</i> )
PBACK	accusé de réception de sonde ( <i>probe acknowledgment</i> )
RD	données de retransmission ( <i>retransmission data</i> )
TC	confirmation de participation à l'arborescence ( <i>tree join confirm</i> )
TCC	confirmation de changement d'arborescence ( <i>tree change confirm</i> )
TCR	demande de changement d'arborescence ( <i>tree change request</i> )
TDC	confirmation de délégation d'arborescence ( <i>tree delegation confirm</i> )
TDR	demande de délégation d'arborescence ( <i>tree delegation request</i> )
TGC	confirmation d'obtention de jeton ( <i>token get confirm</i> )
TGR	demande d'obtention de jeton ( <i>token get request</i> )
TJ	demande de participation à une arborescence ( <i>tree join request</i> )
TLC	confirmation de sortie d'arborescence ( <i>tree leave confirm</i> )
TLR	demande de sortie d'arborescence ( <i>tree leave request</i> )
TNC	confirmation de notification de changement d'arborescence ( <i>tree change notification confirm</i> )
TNR	demande de notification de changement d'arborescence ( <i>tree change notification request</i> )
TRC	confirmation de restitution de jeton ( <i>token return confirm</i> )
TRR	demande de restitution de jeton ( <i>token return request</i> )
TSR	rapport sur l'état des jetons ( <i>token status report</i> )
TSRR	demande de rapport sur l'état des jetons ( <i>token status report request</i> )

## 5 Conventions

Dans la présente Recommandation | Norme internationale, les paquets ECTP-5 sont représentés par des mots en majuscules (par exemple CR pour le paquet de demande de création de connexion) et les paramètres système par des mots en majuscules et en italique (par exemple *TD\_PACKET\_INT* pour l'intervalle entre les paquets de données de test).

## 6 Aperçu général

Le protocole de transport de communications amélioré (ECTP, *enhanced communications transport protocol*) est un protocole de transport visant à prendre en charge les applications multidiffusion Internet. Il fonctionne sur les réseaux IPv4/IPv6 dotés de la capacité de transmission multidiffusion IP au moyen des protocoles de routage IGMP et IP, comme l'indique la Figure 1. Le protocole ECTP peut être configuré en mode UDP.

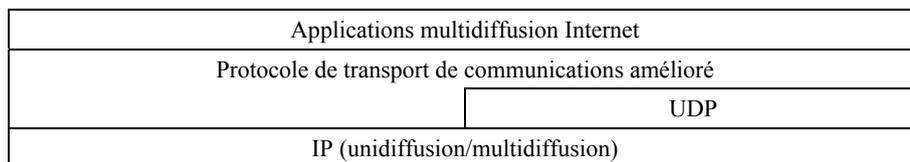


Figure 1 – Modèle de protocole ECTP

La présente Recommandation | Norme internationale décrit la spécification de protocole de la partie 5 du protocole ECTP (ECTP-5) en ce qui concerne la connexion multidiffusée n-plex, en vue de la prise en charge du transport de données multidiffusées entre les participants (utilisateurs du service TS). Dans la connexion multidiffusée n-plex, les utilisateurs du service TS peuvent envoyer des paquets de données multidiffusées au groupe sur la voie de transmission des données multidiffusées. Un utilisateur du service TS qui envoie des données multidiffusées dans la connexion multidiffusée n-plex est un utilisateur expéditeur SU (utilisateur du service TS expéditeur). Tout utilisateur expéditeur SU doit disposer d'un jeton pour la transmission de données multidiffusées. Autrement dit, l'utilisateur du service TS qui obtient un jeton auprès du propriétaire de la connexion TC (TCN) est un utilisateur expéditeur SU.

La Figure 2 indique la voie de transport de données multidiffusées dans la connexion multidiffusée n-plex. Comme le montre la figure, le propriétaire de la connexion TC et l'utilisateur expéditeur SU peuvent transmettre des données multidiffusées aux autres membres de la session à une adresse IP multidiffusion (de groupe).

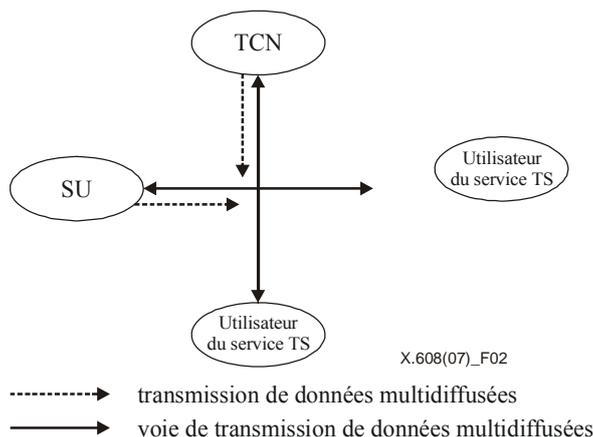


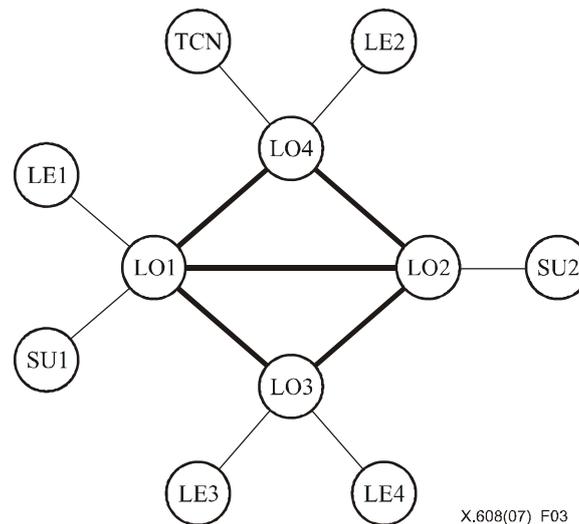
Figure 2 – Transport de données multidiffusées dans une connexion multidiffusée n-plex

Pour établir une connexion multidiffusée n-plex, le propriétaire de la connexion TCN devrait être activé pour gérer les informations de session et les jetons.

Si le propriétaire de la connexion TCN est informé d'une liste de participants par une signalisation hors bande avant le début de la connexion, il devrait lancer une phase de création de connexion en transmettant un paquet CR au groupe. Le paquet CR contient les informations de connexion, y compris les caractéristiques générales de la connexion. Chaque utilisateur du service TS qui figure dans la liste de participants devrait répondre au propriétaire de la connexion TCN par un paquet CC. L'opération de création de connexion s'achève lorsque le propriétaire de la connexion TCN reçoit des paquets CC de tous les utilisateurs du service TS figurant dans la liste des participants, et la phase de transmission de données commence. S'il n'y a aucun participant prédéterminé avant le début de la session, le propriétaire de la connexion TCN lance la phase de transmission de données sans opération de création de connexion.

Au milieu de la phase de transmission de données, les utilisateurs potentiels du service TS devraient se joindre à la connexion en tant que participants tardifs. L'utilisateur du service TS qui est un participant tardif se joint à la connexion en envoyant au propriétaire de la connexion TC un message de demande de participation tardive (JR, *late joint request*). En réponse au message JR, le propriétaire de la connexion TC envoie un message de confirmation de participation tardive (JC, *late joint confirm*) à l'utilisateur du service TS. Une fois confirmée la participation d'un utilisateur du service TS à la session, celui-ci devrait se joindre à une arborescence logique en échangeant avec un propriétaire local (LO) approprié les messages de demande de participation à une arborescence (TJ, *tree join request*) et de confirmation de participation à une arborescence (TC, *tree join confirm*). L'arborescence logique est utilisée pour la protection contre les erreurs.

Une connexion multidiffusée n-plex crée une arborescence logique à deux couches, qui comprend des arborescences intra-groupe partagées et des arborescences inter-groupes par source, comme le montre la Figure 3.



**Figure 3 – Arborescence à deux couches pour la connexion multidiffusée n-plex**

Dans la couche inférieure, chaque entité feuille (LE) d'un groupe local participe à une arborescence logique partagée dont la racine est un propriétaire local (LO) (*arborescence intra-groupe*). Dans la couche supérieure, les propriétaires locaux constituent des arborescences logiques pour chaque utilisateur expéditeur SU (*arborescence inter-groupes*). Il convient de noter que l'on obtient l'arborescence de gestion de chaque utilisateur expéditeur SU en associant ces arborescences inter-groupes et intra-groupe.

Dans une connexion multidiffusée n-plex, les arborescences intra-groupe peuvent être organisées différemment selon le rôle des entités feuilles (LE), qui est déterminé par l'option de configuration d'arborescence (TCO, *tree configuration option*). Une option est que toutes les entités feuilles sont les enfants directs de propriétaires locaux et ne participent pas au processus de réparation et aux regroupements de paquets ACK. Ainsi, dans cette option, l'arborescence intra-groupe sera une arborescence d'un niveau, dont la racine est le propriétaire local (LO). L'autre option est que les entités feuilles sont les enfants d'autres entités feuilles. Dans cette option, les entités feuilles sont chargées d'assurer la fiabilité de leurs enfants et les arborescences intra-groupe peuvent être des arborescences à plusieurs niveaux. Pour que la prise en charge de la fiabilité soit efficace, les arborescences intra-groupe à plusieurs niveaux devraient être aussi proches que possible de l'arborescence de routage multidiffusion sous-jacente. La connexion multidiffusée n-plex applique le mécanisme d'adaptation d'arborescence logique pour cette option. A cet effet, il est fait usage des suites binaires d'erreurs des utilisateurs du service TS. Une suite binaire d'erreurs représente l'état de transmission des paquets, lequel indique la configuration de perte des paquets multidiffusés. Chaque utilisateur du service TS envoie sa suite binaire d'erreurs à son parent concernant les données multidiffusées à partir du nœud racine de son arborescence intra-groupe au moyen de messages ACK périodiques. En comparant ses propres suites binaires d'erreurs et ceux de ses enfants, un nœud décide si chaque enfant est susceptible d'être ou non son enfant effectif dans l'arborescence de routage multidiffusion sous-jacente. S'il est établi que le nœud enfant n'est pas son enfant effectif, le nœud change l'arborescence logique en déléguant le nœud enfant à son parent ou à un de ses autres enfants. Après des changements d'arborescence récurrents, l'arborescence intra-groupe convergera vers une arborescence à plusieurs niveaux qui est plus proche de l'arborescence de routage multidiffusion sous-jacente.

La protection contre les erreurs sera assurée selon l'arborescence de gestion du protocole ECTP qui est créée ainsi qu'il est décrit plus haut. Si la perte de paquets est détectée par un écart entre les numéros de séquence des paquets, un nœud enfant envoie immédiatement un paquet NACK à son parent en mode unidiffusion. Le propriétaire local parent ou l'utilisateur expéditeur SU qui reçoit le paquet NACK retransmettra le paquet de données (RD) au demandeur en mode unidiffusion. Chaque enfant crée un paquet ACK selon la fréquence de création de paquets ACK (*ACK\_GENERATION\_NUM (AGN)*).

Dans la transmission de données multidiffusées, le propriétaire de la connexion TC et les utilisateurs SU peuvent commencer à transmettre des données multidiffusées au groupe au moyen de l'adresse IP multidiffusion et du numéro de port de groupe. Les utilisateurs du service TS livreront les paquets DT reçus à l'application de couche supérieure dans l'ordre indiqué par l'utilisateur expéditeur SU ou le propriétaire de la connexion TC.

Pour le transport de données multidiffusées, un utilisateur du service TS de la connexion peut obtenir un jeton auprès du propriétaire de la connexion TC en envoyant un message TGR. Le propriétaire de la connexion TC répondra alors à l'utilisateur du service TS par le message TGC qui contient un *ID de jeton*. En conséquence, le nombre total de jetons de la connexion est géré par le propriétaire de la connexion TC. L'utilisateur du service TS qui a un jeton est l'utilisateur du service TS expéditeur (SU). Dans certains cas, le propriétaire de la connexion TC peut d'abord demander à un utilisateur

du service TS d'être un utilisateur expéditeur SU en envoyant le message TGR à l'utilisateur expéditeur SU concerné (cession de jeton).

Une fois la transmission de données multidiffusées terminée, l'utilisateur expéditeur SU restituera le jeton au propriétaire de la connexion TC en envoyant un message TRR. Le propriétaire de la connexion TC répondra à l'utilisateur expéditeur SU par un message TRC. Dans certains cas, le propriétaire de la connexion TC demande d'abord à l'utilisateur expéditeur SU de restituer le jeton (retrait de jeton). Le propriétaire de la connexion TC annonce au groupe la situation globale des ID de jetons valides dans la connexion en envoyant les paquets TSR.

Le propriétaire de la connexion TC gère la connexion en ce qui concerne la sortie des utilisateurs. Dans cette opération, un utilisateur du service TS participant peut quitter la connexion en envoyant un message LR au parent. Dans certains cas, le propriétaire de la connexion TC peut obliger un utilisateur du service TS à quitter la connexion en envoyant le message LR (éjection des éléments perturbateurs).

Le propriétaire de la connexion TC peut mettre fin à la connexion n-plex en envoyant un message CT au groupe.

## **7 Considérations**

### **7.1 Participants**

Tous les participants à une connexion multidiffusée n-plex sont des utilisateurs du service TS, l'un d'entre eux étant le propriétaire de la connexion TC (TCN).

Propriétaire de la connexion TC (TCN, *TC-owner*):

Une connexion multidiffusée n-plex a un seul propriétaire TCN, qui est chargé de la gestion de la connexion, notamment de la création et de la fin de la connexion, de la mise à jour de la connexion et de la gestion des jetons.

Ainsi, dans les applications de téléconférence, le propriétaire de la connexion TC peut agir en qualité de "serveur de conférence", serveur qui peut être utilisé pour la gestion de la conférence sans envoi de données multidiffusées. Dans l'exemple des "jeux en ligne à multiples utilisateurs", le propriétaire de la connexion TC peut agir en tant que "serveur de gestion de jeux".

Utilisateur du service TS (utilisateur du service de transport):

Une connexion multidiffusée n-plex comprend un ou plusieurs utilisateurs du service TS, chacun d'entre eux envoyant et recevant des données multidiffusées dans la connexion.

Un utilisateur du service TS peut devenir un propriétaire local (LO) ou une entité feuille (LE) selon le rôle qu'il assume.

Propriétaire local (LO, *local owner*):

Un propriétaire local est un nœud représentatif d'un groupe local désigné de manière statique. Il est chargé de mettre à jour une arborescence intra-groupe du groupe et les arborescences de gestion pour tous les utilisateurs SU de son groupe local. Chaque propriétaire local est aussi connecté aux autres propriétaires locaux au moyen d'arborescences inter-groupes. Périodiquement, il crée aussi un trafic d'essai pour l'adaptation des arborescences logiques.

Entité feuille (LE, *leaf entity*):

Une entité feuille est un membre d'un groupe local dont les représentants sont un propriétaire local (LO). Elle devrait participer à une arborescence intra-groupe du groupe et elle est chargée d'échanger des paquets de gestion avec les entités feuilles parents ou enfants le long de l'arborescence de gestion.

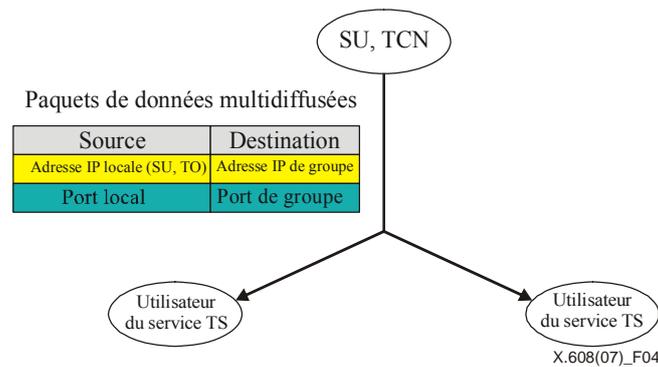
Un utilisateur du service TS peut devenir un utilisateur expéditeur SU lorsqu'il obtient un jeton du propriétaire de la connexion TCN.

Utilisateur expéditeur du service TS (SU, *sending TS-user*):

Un utilisateur expéditeur SU est un utilisateur du service TS qui peut envoyer des données multidiffusées au groupe. Dans une connexion multidiffusée n-plex, un utilisateur du service TS devient un utilisateur expéditeur SU lorsqu'il a un jeton et qu'il peut donc transmettre des données multidiffusées au groupe.

### **7.2 Voie de transmission de données et adressage**

Dans une connexion multidiffusée n-plex, l'utilisateur expéditeur SU ou le propriétaire de la connexion TC peut envoyer des paquets de données multidiffusées aux membres de la session comme l'indique la Figure 4.



**Figure 4 – Adressage des données multidiffusées dans une connexion multidiffusée n-plex**

Dans une connexion multidiffusée n-plex, les paquets de données multidiffusées (DT) utilisent l'adresse IP multidiffusion et le numéro de port de groupe comme adresse de destination. L'adresse source des paquets de données multidiffusées IP est l'adresse IP de l'expéditeur du paquet. Par contre, les paquets de données de retransmission (RD) envoyés en réponse aux délégations de réparations (NACK) sont livrés par le propriétaire local ou l'utilisateur expéditeur SU sur la voie de gestion unidiffusion, l'adresse IP du demandeur de la réparation étant la destination.

### 7.3 Voie et arborescence de gestion

Dans une connexion multidiffusée n-plex, des voies de gestion sont consacrées aux reprises après incident. Tous les membres participent à une ou plusieurs arborescences de gestion. Celles-ci servent de voies de gestion pour l'échange de messages de gestion entre participants. Un nœud parent agit en tant qu'agent pour aider les nœuds enfants dans les reprises après perte de paquets. Il regroupe aussi les informations d'accusé de réception provenant des nœuds descendants. Tous les paquets de gestion tels que les paquets RD, NACK et ACK sont transmis au moyen de la voie de gestion unidiffusion.

Une connexion multidiffusée n-plex est divisée en plusieurs groupes locaux de participants aux fins de la protection contre les erreurs. Selon cette approche de groupe, les participants créent et mettent à jour des arborescences de gestion le long desquelles tous les paquets de protection contre les erreurs sont échangés. Les arborescences de gestion sont construites à partir de deux couches d'arborescences logiques. Dans la couche inférieure, les membres d'un groupe local participent à une arborescence logique partagée dont la racine est un propriétaire local (LO) (*arborescence intra-groupe*). Dans la couche supérieure, les propriétaires locaux des groupes constituent des arborescences logiques par source (*arborescence inter-groupes*). Autrement dit, chaque entité feuille (LE) participe à un groupe local et elle se greffe sur l'arborescence logique du groupe dont la racine est un propriétaire local (LO), en tant qu'enfant (ou descendant) du propriétaire local. Chaque propriétaire local se greffe sur l'arborescence logique des propriétaires locaux, dont la racine est le propriétaire local du groupe auquel l'utilisateur expéditeur SU appartient. Une arborescence de gestion pour chaque source est créée en connectant les arborescences inter-groupes et les arborescences intra-groupe.

Autant d'arborescences inter-groupes sont créées et mises à jour qu'il est nécessaire. Toute arborescence inter-groupes plus importante dont la racine est un propriétaire local qui n'a pas d'utilisateur expéditeur SU comme enfant devrait être supprimée.

Les arborescences intra-groupe peuvent devenir des arborescences à plusieurs niveaux qui prennent en compte les chemins de routage multidiffusion effectifs grâce au mécanisme d'adaptation des arborescences logiques décrit au § 7.5.

En traversant les arborescences logiques à partir d'un utilisateur expéditeur SU, on peut obtenir une arborescence dont la racine est l'utilisateur expéditeur SU et qui est l'arborescence de gestion de cet utilisateur.

Si un utilisateur expéditeur SU appartient à un autre groupe local, une partie de l'arborescence de gestion créée pour le groupe local existant est l'arborescence intra-groupe du groupe. Les arborescences de gestion des utilisateurs SU appartenant au même groupe local sont légèrement différentes de l'arborescence intra-groupe, car les nœuds intermédiaires entre le propriétaire local et l'utilisateur expéditeur SU ont inversé la relation parent-enfant.

Si les arborescences logiques des participants sont construites selon la Figure 5, l'arborescence de gestion de la source SU1 le sera comme l'indique la Figure 6.

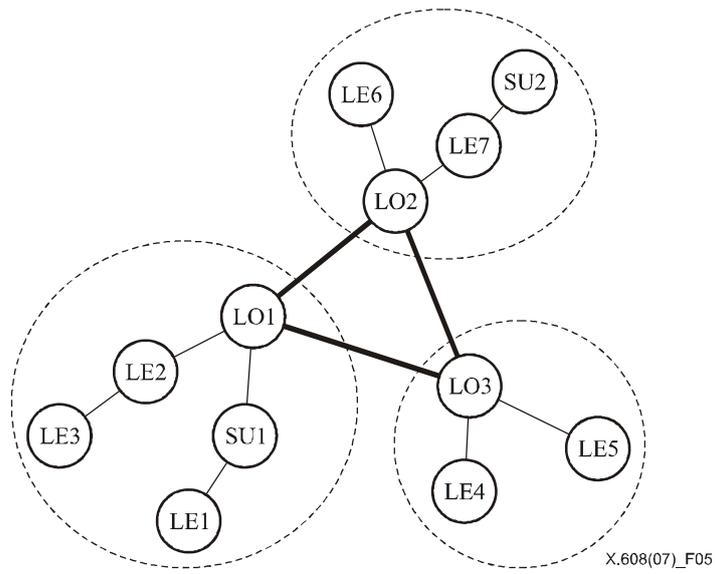


Figure 5 – Deux couches d'arborescences logiques

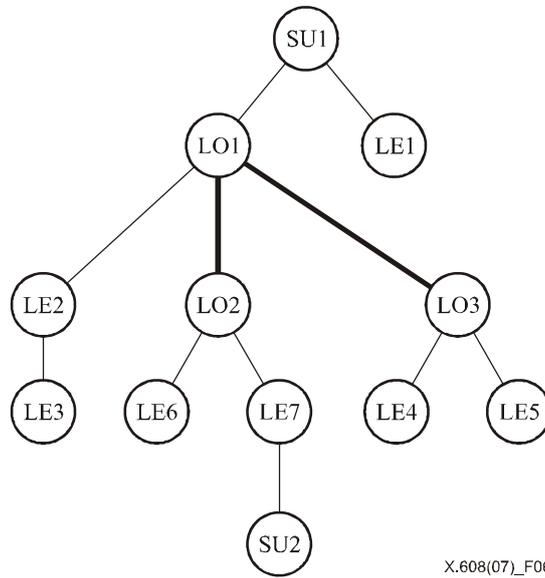
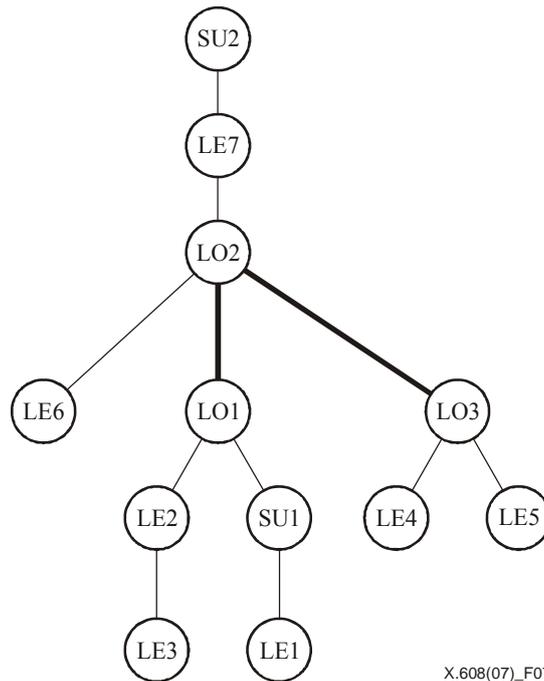


Figure 6 – Arborescence de gestion avec l'expéditeur SU1

De même, la Figure 7 représente l'arborescence de gestion dont la source est SU2.



**Figure 7 – Arborescence de gestion avec l'expéditeur SU2**

Les arborescences de gestion sont mises à jour au moyen des informations obtenues grâce à l'échange de paquets NACK et ACK. Le parent peut détecter la défaillance d'un enfant en comparant son numéro de séquence le plus bas (LSN) à celui reçu de l'enfant. Les enfants peuvent détecter la défaillance de leur parent en l'absence de réponse du parent à plusieurs demandes de réparation consécutives. L'enfant trouve ensuite un autre parent en contactant le propriétaire local (LO).

#### 7.4 Jetons

Dans une connexion multidiffusée n-plex, un jeton représente le droit d'un utilisateur du service TS d'envoyer des données multidiffusées au propriétaire de la connexion TC. Avant de transmettre des données, chaque utilisateur du service TS doit obtenir un jeton auprès du propriétaire de la connexion TC, selon les procédures de gestion de jetons utilisées dans la connexion multidiffusée n-plex. Ainsi, le propriétaire de la connexion TC peut autoriser un utilisateur du service TS à devenir un expéditeur, de sorte que les utilisateurs du service TS peuvent effectivement filtrer et écarter les données multidiffusées envoyées par des utilisateurs non autorisés. Cela étant, il convient de noter que l'utilisation de jetons n'offre aucune protection pour la multidiffusion IP.

Chaque jeton est représenté par un entier non négatif de 1 octet. Ce numéro de jeton (ou ID de jeton) sera attribué par le propriétaire de la connexion TC lorsqu'un utilisateur du service TS demande un jeton dans la connexion. L'ID de jeton a une valeur comprise entre 1 et 255, la valeur '0' étant réservée à l'usage du propriétaire de la connexion TC. Du côté du destinataire, l'ID de jeton peut servir à déterminer les entités qui peuvent envoyer des données multidiffusées.

#### 7.5 Adaptation des arborescences logiques

Dans une connexion multidiffusée n-plex, les arborescences intra-groupe peuvent devenir des arborescences à plusieurs niveaux proches des arborescences de routage multidiffusion sous-jacentes lorsque l'option de configuration d'arborescence TCO a la valeur '10'. La comparaison des configurations de perte permet d'estimer les arborescences de routage multidiffusion sous-jacentes. L'estimation est réalisable étant donné que si une perte survient à un nœud parent d'une arborescence de routage multidiffusion, tous ses enfants subissent la même perte. La racine d'une arborescence intra-groupe ne change pas suite au processus d'adaptation des arborescences logiques.

Dans ce processus, chaque destinataire d'une session multidiffusion tient à jour l'état de transmission des paquets, appelé suite binaire d'erreurs, qui indique la configuration de perte des paquets multidiffusés. Une telle suite binaire comprend deux parties, le numéro de séquence (*Ns*) et la suite binaire elle-même (*B*). *Ns* est le numéro de séquence du premier paquet d'une séquence de paquets représentés dans la suite binaire. Un bit de *B* indique l'état de réception du paquet correspondant, '1' signifiant que les paquets ont pu être transmis à un destinataire sans reprise après incident

et '0' signifiant une défaillance. La suite binaire comprend les configurations de perte des paquets multidiffusés. Ainsi, si  $N_s = 5$  et  $B = 11010$ , le destinataire a bien reçu les paquets 5, 6 et 8. Si les paquets 7 et 9 ont été récupérés grâce aux retransmissions, ces bits sont à '0'. Chaque destinataire envoie périodiquement la suite binaire d'erreurs à son parent dans l'arborescence. Si une arborescence logique correspond à une arborescence de routage multidiffusion et si un bit de la suite binaire à un nœud a la valeur '1', le bit correspondant de la suite binaire de son parent sera probablement à '1'. Il convient de noter que tous les éléments de la suite binaire d'erreurs d'une source ont toujours la valeur '1'.

Chaque nœud qui a des nœuds enfants compare sa propre suite binaire d'erreurs et celle de ses enfants pour vérifier si la relation entre lui et ses enfants correspond étroitement à l'arborescence de routage multidiffusion sous-jacente. Un nœud parent peut déduire que certains nœuds enfants doivent changer d'emplacement dans l'arborescence logique. Le nœud enfant est un enfant d'un autre nœud enfant, sinon il n'est pas un descendant du nœud parent.

Par exemple, la Figure 8 donne un exemple d'arborescence de routage multidiffusion et de suite binaire d'erreurs de chaque nœud.

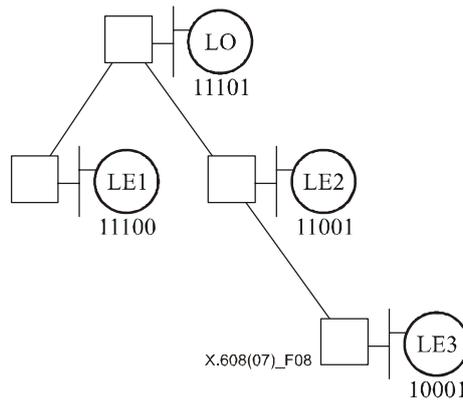


Figure 8 – Exemple d'arborescence de routage multidiffusion

On trouvera aux Figures 9 et 10 une description succincte de la façon dont une arborescence logique proche d'une arborescence de routage multidiffusion est construite selon les informations données par la suite binaire d'erreurs.

Premièrement, on suppose qu'un propriétaire local racine est un serveur déployé de façon stratégique qui est habituellement situé près du point de sortie d'un fournisseur de services Internet. Les carrés représentent les routeurs et les numéros figurant au-dessous des nœuds sont les suites binaires d'erreurs de chaque nœud.

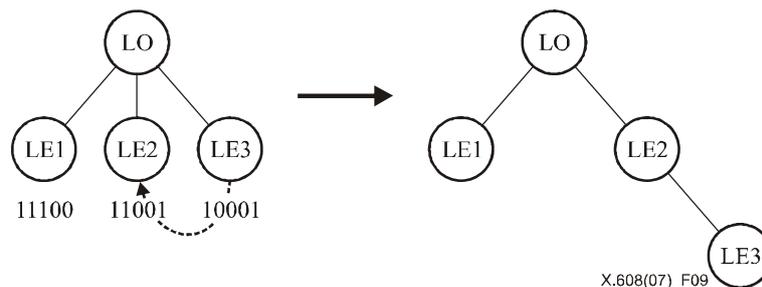
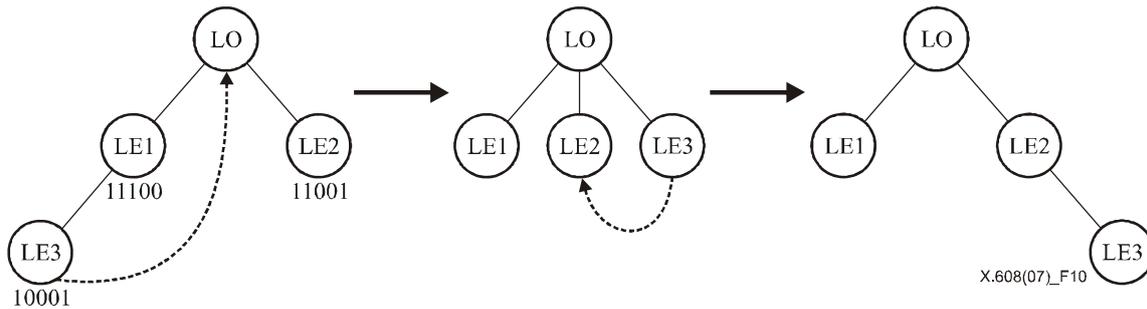


Figure 9 – Adaptation des arborescences logiques (LE2 adopte LE3)

La partie gauche de la Figure 9 représente une arborescence intra-groupe de nœuds à un niveau initiale, qui est reproduite à la Figure 8. Lorsque le propriétaire local prend connaissance des suites binaires d'erreurs de ses enfants, il découvre que l'entité LE3 est probablement un enfant ou un descendant de l'entité LE2 puisqu'un ensemble de paquets reçus de l'entité LE3 est un sous-ensemble de celui de l'entité LE2. Ainsi, il délègue l'entité LE3 à l'entité LE2 en envoyant un message TDR (demande de délégation d'arborescence). À réception du message, l'entité LE2 estime que l'entité LE3 est son enfant et envoie un message TCR (demande de changement d'arborescence) à l'entité LE3. À réception du message TCR, l'entité LE3 devient participant en tant qu'enfant de l'entité LE2 en envoyant le message TJ à l'entité LE2 et quitte son parent précédent, LO, en envoyant un message TLR. Il est à noter qu'un nœud qui reçoit un message TCR est rattaché au nouveau parent avant de quitter son parent précédent. Il en résulte une arborescence dans la partie droite de la Figure 9.

On trouvera à la Figure 10 un autre exemple d'adaptation d'arborescence logique.



**Figure 10 – Adaptation d'arborescence logique (LO adopte LE3; LE2 adopte LE3)**

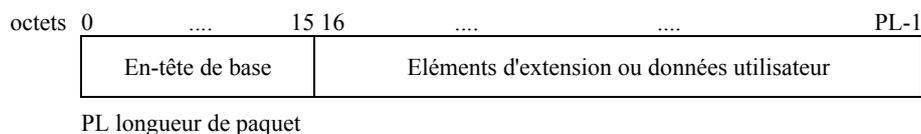
Dans la partie gauche de la Figure 10, l'entité LE3 est devenue par erreur un enfant de l'entité LE1 en raison de l'adaptation d'arborescence précédente. Toutefois, une suite binaire d'erreurs existante de l'entité LE3 indique que cette dernière n'est pas un enfant de l'entité LE1 puisqu'elle a reçu un paquet qui n'est pas reçu par l'entité LE1. Dans ce cas, l'entité LE1 délègue l'entité LE3 à son parent, le propriétaire local LO, pour qu'il trouve la position appropriée de l'entité LE3 en envoyant un message TDR. Le propriétaire local reçoit le message TDR de l'entité LE1 concernant l'entité LE3 et compare la suite binaire d'erreurs de l'entité LE3 à celle de ses enfants LE1 et LE2. Il découvre que l'entité LE3 est probablement un enfant de l'entité LE2. Enfin, l'arborescence logique converge vers l'arborescence représentée dans la partie droite de la Figure 10.

Ainsi, les arborescences intra-groupe peuvent devenir des arborescences à plusieurs niveaux proches des arborescences de routage multidiffusion sous-jacentes. Les arborescences de gestion créées à partir de ces arborescences intra-groupe seront aussi très proches des arborescences de routage multidiffusion.

Le propriétaire local envoie les paquets de données de test lorsqu'il détecte un changement dans l'arborescence intra-groupe. Les entités feuilles du groupe local créent ensuite les informations relatives à la suite binaire d'erreurs et les envoient au moyen du paquet ACK. Les paquets de données de test ne sont destinés à aucune application et devraient être envoyés à chaque intervalle d'envoi de paquets de données de test (*TD\_PACKET\_INT*) et un nombre de fois égal au nombre de paquets de données de test (*TD\_PACKET\_NUM*).

## 8 Paquets

Un paquet ECTP contient un en-tête de base de 16 octets ainsi que des éléments d'extension ou des données utilisateur. Il convient de noter que les paquets de données (DT) ne comprennent pas d'éléments d'extension. Le format de paquet général du protocole ECTP-5 est indiqué dans la Figure 11.



**Figure 11 – Format de paquet du protocole ECTP-5**

### 8.1 En-tête de base

L'en-tête de base de 16 octets contient des informations communes utiles pour toutes les opérations protocolaires, en particulier pour les paquets de données. La Figure 12 représente la structure de l'en-tête de base lorsque le protocole ECTP fonctionne sur IP.

0				1				2				3											
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Elément suivant				Version		CT		Type de paquet				Total de contrôle											
Port source								Port de destination															
Numéro de séquence de paquet (PSN)																							
Longueur de capacité utile								F	Réservé				ID de jeton										

Figure 12 – En-tête de base (ECTP sur IP)

L'en-tête de base contient les informations suivantes:

*Elément suivant* (4 bits)

Indique le type d'élément d'extension suivant immédiatement l'en-tête de base. Les valeurs de codage des éléments d'extension seront décrites ultérieurement. La valeur '0000' signifie qu'aucun autre élément d'extension ne vient après cet élément.

*Version* (2 bits)

Définit la version du protocole pour la présente Recommandation | Norme internationale. La version en cours est codée à '00'.

*CT (type de connexion)* (2 bits)

Spécifie le type de connexion ECTP. Les valeurs de codage sont les suivantes:

01 – connexion multidiffusée simplex (pour les protocoles ECTP-1 et ECTP-2);

10 – connexion multidiffusée duplex (pour les protocoles ECTP-3 et ECTP-4);

11 – connexion multidiffusée n-plex (pour les protocoles ECTP-5 et ECTP-6).

Dans la présente Recommandation | Norme internationale, le type *CT* doit être mis à '11'.

*Type de paquet* (8 bits)

Indique le type de paquet.

*Total de contrôle* (16 bits)

Utilisé pour vérifier la validité du paquet qui comprend l'en-tête de base, l'en-tête d'extension et/ou les données utilisateur. Le total de contrôle est calculé au moyen de l'opération arithmétique classique du complément à un, comme c'est le cas avec les protocoles TCP et UDP.

*Port source* (16 bits) et *Port de destination* (16 bits)

Ces numéros de port permettent d'identifier les applications expéditrices et destinataires dans le cas du protocole ECTP sur IP. Lorsque le protocole ECTP fonctionne avec le protocole UDP, ces champs servent à représenter l'ID de connexion, comme on le verra ultérieurement.

*PSN (Numéro de séquence de paquet)* (32 bits)

Pour les paquets DT, ce champ est un nombre non signé de 32 bits qui commence par le numéro de séquence initial, augmente de '1' à chaque paquet et reboucle sur '1' après avoir atteint la valeur  $2^{32} - 1$ .

Pour les paquets ACK, ce champ est le numéro de séquence le plus bas (LSN) d'un utilisateur expéditeur SU ayant un *ID de jeton*.

Pour les paquets RD, ce champ est le numéro PSN du paquet de données dont la retransmission est demandée.

Pour les paquets TJ, TLR, JR, TGR, TRR, TCR, TDR, TNR et CCR, ce champ est le numéro de séquence des paquets de gestion provenant d'un nœud. Il devrait être unique dans le même type de paquet envoyé par le nœud.

Pour les paquets TC, TLC, JC, TGC, TRC, TCC, TDC, TNC et CCC, ce champ est calqué sur le champ *PSN* du paquet de demande correspondant.

Pour les autres paquets, ce champ devrait être ignoré.

*Longueur de capacité utile* (16 bits)

Cette valeur indique la longueur totale, en octets, des en-têtes d'extension ou des données utilisateur suivant l'en-tête de base.

*F* (1 bit)

Il s'agit d'un bit indicateur, dont l'utilisation dépend du type de paquet:

Pour les paquets JC, TC, TGC, TCC, TDC, TRC et TLC, la valeur  $F = 1$  indique que chaque demande de participation correspondante est acceptée. Dans le cas contraire,  $F$  est mis à '0'.

Pour les paquets TJ et TLR,  $F$  est mis à '1' pour la participation à l'arborescence inter-groupes ou la sortie de cette arborescence ou à '0' pour la participation à l'arborescence intra-groupe ou la sortie de cette arborescence.

Pour le paquet LR,  $F$  est mis à '1' pour une sortie demandée par l'utilisateur ou à '0' pour l'éjection des éléments perturbateurs.

Pour le paquet CT,  $F$  est mis à '1' pour une fin anormale ou à '0' pour une fin normale après la transmission de toutes les données.

Pour les opérations de gestion de jetons, les messages de demande TGR et TRR se servent de cet indicateur pour signaler s'il s'agit d'une gestion de jeton initiée par le propriétaire de la connexion TC (TCN) ou par l'utilisateur du service TS.

Pour le paquet TSR, la valeur  $F = 1$  indique que l'état global des jetons est modifié suite à l'adjonction d'un nouveau jeton ou à la suppression d'un jeton existant. En revanche, la valeur  $F = 0$  signifie que le paquet TSR a une valeur d'information.

Pour le paquet TNR,  $F$  est mis à '1' pour une participation à l'arborescence ou '0' pour une sortie de l'arborescence.

Pour le paquet DT,  $F$  est mis à '1' pour les données de test provenant du propriétaire local en vue de l'adaptation de l'arborescence logique ou à '0' pour les données émanant d'applications normales.

Pour le paquet RD,  $F$  est mis à '1' si l'expéditeur n'a pas de données à réparer, sinon il est mis à '0'.

Pour les autres paquets, ce champ ne sera pas pris en compte.

*Réservé* (7 bits)*ID de jeton* (8 bits)

L'*ID de jeton* est valide uniquement pour les paquets DT et RD. Il représente la source du paquet de données. Sa valeur est comprise entre 0 et 255. Chaque utilisateur expéditeur SU reçoit un *ID de jeton* du propriétaire de la connexion TC au moyen de la procédure d'obtention et de cession de jeton et affecte à ce champ le numéro attribué par le propriétaire de la connexion TC.

Par ailleurs, lorsque le protocole ECTP fonctionne avec le protocole UDP, il n'est pas nécessaire que l'en-tête de paquet précise les ports source et de destination, qui seront indiqués par l'en-tête du protocole UDP. Dans ce cas, le champ de 32 bits correspondant aux ports source et de destination sera rempli par "*ID de connexion*". Par défaut, il peut recevoir la valeur de l'adresse de groupe IPv4.

Le format de l'en-tête de base pour le protocole ECTP sur UDP est décrit par la Figure 13:

0							1							2							3													
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7			
Elément suivant							Version							CT							Type de paquet							Total de contrôle						
ID de connexion																																		
Numéro de séquence de paquet																																		
Longueur de capacité utile														F	Réservé							ID de jeton												

**Figure 13 – En-tête de base (ECTP sur UDP)**

L'*ID de connexion* permet à l'hôte ECTP d'identifier une connexion ECTP. Il peut aussi servir à vérifier la connexion. Dans la phase d'établissement de la connexion, cette information doit d'abord être communiquée par le propriétaire de la connexion TC aux autres participants au moyen des paquets CR ou JC. Tous les autres paquets décrits dans la présente Recommandation | Norme internationale doivent avoir, pour ce champ, la valeur indiquée par le propriétaire de la connexion TC.

## 8.2 Eléments d'extension

Les paquets de gestion ECTP peuvent contenir un ou plusieurs éléments d'extension conjointement avec l'en-tête de base. L'en-tête de base et chaque élément d'extension ont un champ Élément suivant qui pointe sur l'élément d'extension qui vient éventuellement immédiatement après.

Le champ Élément suivant est codé comme l'indique le tableau ci-après. Le dernier élément d'extension d'un paquet doit mettre son champ En-tête suivant à la valeur '0000' (aucun élément).

Tableau 1 – Eléments d'extension

Elément d'extension	Valeur de codage de l'élément suivant de l'élément précédent (4 bits)	Longueur de l'élément d'extension (octets)
Aucun élément	0000	0
Connexion	0001	4
Suite binaire d'erreurs	0010	Variable
Horodateur	0100	12
Jeton	0110	Variable
Informations sur le propriétaire local	0111	Variable
Accusé de réception négatif	1000	8
Information de changement d'arborescence	1001	8

### 8.2.1 Élément Connexion

L'élément d'extension Connexion contient des informations globales sur la connexion de transport. Il est codé à '0001' dans le champ Élément suivant de l'élément précédent ou de l'en-tête de base. Il doit être inclus dans les paquets CR, JC et TGR. Sa structure de l'élément est indiquée dans la Figure 14, sur une longueur de '4' octets:

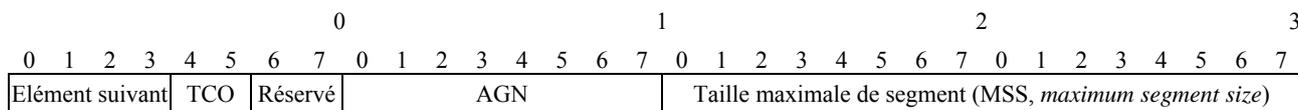


Figure 14 – Élément d'extension Connexion

Chaque champ est défini comme suit:

*Elément suivant* (4 bits)

Précise le type d'élément d'extension suivant, comme l'indique le Tableau 1.

*TCO (option de configuration d'arborescence)* (2 bits)

Précise l'option de configuration d'arborescence utilisée dans le protocole ECTP-5 comme suit:

00 – Réservé pour une future utilisation.

01 – Arborescence intra-groupe à un niveau sans adaptation d'arborescence logique.

10 – Arborescence intra-groupe à plusieurs niveaux avec adaptation d'arborescence.

11 – Réservé pour une future utilisation.

La valeur par défaut de l'option *TCO* est '10' dans une connexion multidiffusée n-plex.

*Réservé* (2bits)

*AGN (fréquence de création de paquets ACK)* (8 bits)

Entier positif compris entre 1 et 255 (*ACK\_GENERATION\_NUM*). La fréquence AGN est utilisée par un utilisateur du service TS enfant pour créer et transmettre un paquet ACK au parent.

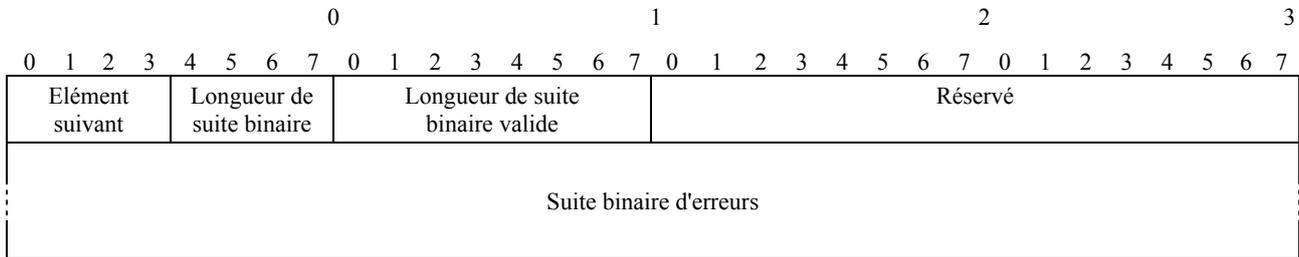
*MSS* (16 bits)

Définit la taille maximale (en octets) du segment de données utilisateur (*MAX\_SEGMENT\_SIZE*) susceptible d'être contenu dans le paquet DT. La valeur par défaut est '1024'.

**8.2.2 Elément Suite binaire d'erreurs**

Cet élément d'extension donne des informations sur l'état de réception des paquets à un nœud enfant. Il est rattaché au paquet ACK en réponse au trafic de test du propriétaire local (LO) en vue de l'adaptation d'arborescence logique décrite au § 7.5. Il est codé à '0010' dans le champ Elément suivant de l'élément précédent ou de l'en-tête de base.

Cet élément comprend quatre octets fixes et la *suite binaire d'erreurs* de longueur variable, ainsi qu'il est décrit dans la Figure 15.



**Figure 15 – Elément d'extension Suite binaire d'erreurs**

Chaque champ est défini comme suit:

*Elément suivant* (4 bits)

Précise le type d'élément d'extension suivant, comme l'indique le Tableau 1.

*Longueur de suite binaire* (4 bits)

Définit la taille totale de la variable *Suite binaire du paquet ACK* en unités de mot (4 octets).

*Longueur de suite binaire valide* (8 bits)

Représente en "bits" la longueur de la partie réellement valide de la variable *Suite binaire du paquet ACK*.

*Réservé* (16 bits)

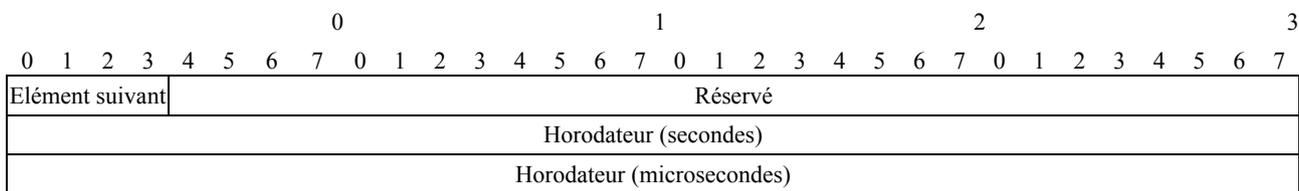
*Suite binaire d'erreurs* (variable)

Représente par '0' ou '1' les informations indiquant quels paquets de données ont été reçus (1) ou perdus (0) par le destinataire. Dans la *suite binaire d'erreurs*, les informations commencent par le numéro de séquence *LSN*, qui représente le numéro de séquence du paquet DT au numéro est le plus bas qui n'a pas encore été reçu par le destinataire. Le numéro *LSN* sera défini dans le champ *PSN* de l'en-tête de base. La *suite binaire d'erreurs* contient les informations totales concernant la taille de la variable *Longueur de suite binaire valide*.

**8.2.3 Elément Horodateur**

L'élément Horodateur est codé à '0100 ' dans le champ Elément suivant de l'élément précédent ou de l'en-tête de base. Le protocole ECTP-5 utilise un horodateur de 8 octets pour calculer le temps de transmission aller retour (RTT, *round trip time*).

L'élément d'extension Horodateur de 12 octets a le format suivant, comme le montre la Figure 16:



**Figure 16 – Elément d'extension Horodateur**

Chaque champ est défini comme suit:

*Elément suivant* (4 bits)

Précise le type d'élément d'extension suivant, comme l'indique le Tableau 1.

*Réservé* (28 bits)

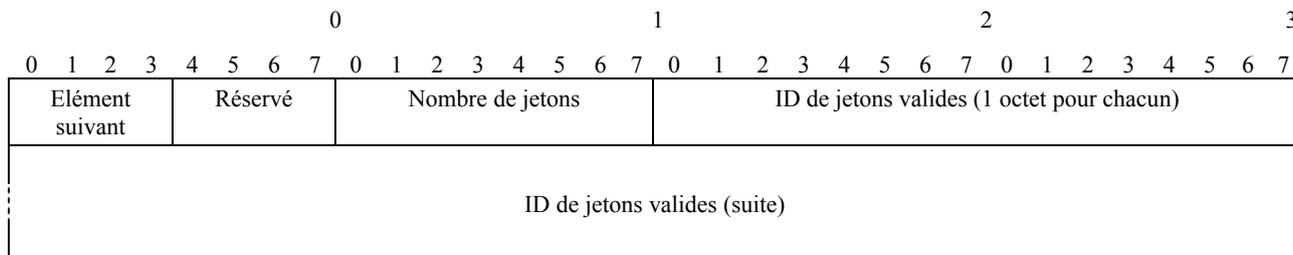
*Horodateur* (64 bits)

Contient la valeur de l'horodateur de 8 octets. Les quatre premiers octets représentent la valeur temporelle en secondes, les quatre octets suivants en microsecondes, comme le fait le programme *ping* classique.

**8.2.4 Elément Jeton**

Cet élément d'extension donne des informations sur l'état des jetons utilisés dans la connexion. Cet en-tête d'extension est rattaché au paquet TSR. Il est codé à '0110' dans le champ Elément suivant de l'élément précédent ou de l'en-tête de base.

Il comprend deux octets fixes et les *ID de jetons valides* de longueur variable, ainsi qu'il est décrit dans la Figure 17:



**Figure 17 – Elément d'extension Jeton**

Chaque champ est défini comme suit:

*Elément suivant* (4 bits)

Précise le type d'élément d'extension suivant, comme l'indique le Tableau 1.

*Réservé* (4 bits)

*Nombre de jetons* (8 bits)

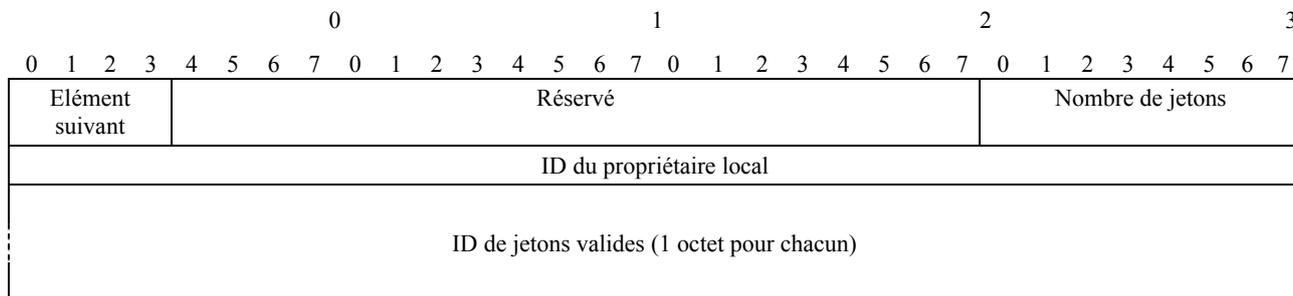
Précise le nombre total d'*ID de jetons valides* dans la connexion.

*ID de jetons valides* (variable)

Contient la liste des ID de jetons valides dans la connexion. La longueur de chaque *ID de jeton* est de 1 octet.

**8.2.5 Elément Informations sur le propriétaire local**

Cet élément d'extension donne des informations sur un propriétaire local (LO) et les ID de jetons correspondants. Avec ces informations, un propriétaire local peut reconnaître le propriétaire local parent le long de l'arborescence inter-groupes menant vers un utilisateur expéditeur SU. Cet en-tête d'extension est joint aux paquets TSR, TGR et TGC. Il est codé à '0111' dans le champ Elément suivant de l'élément précédent ou de l'en-tête de base. Il comprend huit octets fixes et les ID de jetons correspondants de longueur variable, ainsi qu'il est décrit dans la Figure 18.



**Figure 18 – Elément d'extension Informations sur le propriétaire local**

Chaque champ est défini comme suit:

*Elément suivant* (4 bits)

Précise le type d'élément d'extension suivant, comme l'indique le Tableau 1.

*Réservé* (20 bits)

*Nombre de jetons* (8 bits)

Précise le nombre total d'*ID de jetons correspondants* pour le propriétaire local dont l'*ID* est l'*ID du propriétaire local*.

*ID du propriétaire local* (32 bits)

Indique l'*ID* du propriétaire local auquel se joignent les utilisateurs SU des *ID de jetons correspondants*. Cet *ID* peut être attribué pour une application particulière.

*ID de jetons correspondants* (variable)

Contient la liste des *ID* de jetons correspondant à l'*ID du propriétaire local* dans la connexion. La longueur de chaque *ID de jeton* est de un octet.

### 8.2.6 Élément Accusé de réception négatif

Cet élément d'extension donne des informations sur une demande de réparation concernant les paquets DT perdus à un nœud enfant. Si une entité feuille (LE) ou un propriétaire local détecte une ou plusieurs pertes de paquets DT, il demande la réparation du ou des paquets perdus à son parent LE ou propriétaire local le long de l'arborescence de gestion. Si plusieurs paquets consécutifs sont détectés comme étant perdus, l'entité feuille ou le propriétaire local peut demander la réparation d'un bloc de paquets perdus au moyen d'un seul paquet NACK qui contient le numéro de séquence de départ et le nombre de paquets consécutifs perdus.

Cet en-tête d'extension est rattaché au paquet NACK. Il est codé à '1000' dans le champ Élément suivant de l'élément précédent de l'en-tête de base. Il comprend huit octets fixes ainsi qu'il est décrit dans la Figure 19:

	0							1							2							3									
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Elément suivant				Réservé												Nombre de paquets perdus															
Numéro de séquence du paquet de départ																															

Figure 19 – Élément d'extension Accusé de réception négatif

Chaque champ est défini comme suit:

*Elément suivant* (4 bits)

Précise le type d'élément d'extension suivant, comme l'indique le Tableau 1.

*Réservé* (12 bits)

*Nombre de paquets perdus* (16 bits)

Indique le nombre de paquets consécutifs perdus à partir du *Numéro de séquence du paquet de départ*. Si la demande vise la perte d'un seul paquet, ce champ est mis à '1'.

*Numéro de séquence du paquet de départ* (32 bits)

Précise le numéro de séquence du paquet de départ d'un bloc de paquets perdus. Si la demande vise la perte d'un seul paquet, ce champ est mis à la valeur PSN du paquet perdu.

Un paquet NACK est créé pour les paquets de données d'un utilisateur expéditeur SU. Le champ *ID de jeton* de l'en-tête de base du paquet NACK identifie l'expéditeur associé qui a envoyé les paquets perdus.

### 8.2.7 Élément Informations de changement d'arborescence

Cet élément d'extension donne des informations sur un changement d'arborescence intra-groupe. Il représente un changement d'arborescence intra-groupe concernant l'expéditeur et un nœud doté de l'*ID de nœud* en question. Cet en-tête d'extension est joint aux paquets TCR, TDR, TNR et CCR. Il est codé à '1001' dans le champ Élément suivant de l'élément précédent ou de l'en-tête de base. Il comprend huit octets fixes.

	0							1							2							3									
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Elément suivant				Réservé																											
ID de nœud																															

Figure 20 – Élément d'extension Informations de changement d'arborescence

## ISO/CEI 14476-5:2008 (F)

Chaque champ est défini comme suit:

*Élément suivant* (4 bits)

Précise le type d'élément d'extension suivant, comme l'indique le Tableau 1.

*Réservé* (28 bits)

*ID de nœud* (32 bits)

Indique l'ID d'un nœud pour lequel l'arborescence intra-groupe devrait être changée. Le contexte du changement dépend du type de paquet.

### 8.3 Format de paquet

Dans la présente Recommandation | Norme internationale, il existe au total 30 types de paquets: deux types de paquets de données et 28 types de paquets de gestion. Les paquets de données sont des paquets DT et RD.

**Tableau 2 – Paquets du protocole ECTP-5**

Nom entier	Abréviation	Transport	De	A
Demande de création de connexion	CR	Multidiffusé	Propriétaire de la connexion TC	Utilisateurs du service TS
Confirmation de création de connexion	CC	Unidiffusé	Utilisateur du service TS	Propriétaire de la connexion TC
Demande de participation à une arborescence	TJ	Unidiffusé	Entité feuille/Propriétaire local	Entité feuille/Propriétaire local
Confirmation de participation à une arborescence	TC	Unidiffusé	Entité feuille/Propriétaire local	Entité feuille/Propriétaire local
Demande de sortie d'arborescence	TLR	Unidiffusé	Entité feuille/Propriétaire local	Entité feuille/Propriétaire local
Confirmation de sortie d'arborescence	TLC	Unidiffusé	Entité feuille/Propriétaire local	Entité feuille/Propriétaire local
Données	DT	Multidiffusé	Propriétaire de la connexion TC/Utilisateur expéditeur SU/Propriétaire local	Utilisateurs du service TS
Données de retransmission	RD	Unidiffusé	Entité feuille/Propriétaire local	Entité feuille/Propriétaire local
Accusé de réception	ACK	Unidiffusé	Entité feuille/Propriétaire local	Entité feuille/Propriétaire local
Accusé de réception négatif	NACK	Unidiffusé	Entité feuille/Propriétaire local	Entité feuille/Propriétaire local
Sonde	PB	Unidiffusé	Propriétaire de la connexion TC	Utilisateur du service TS
Accusé de réception de sonde	PBACK	Unidiffusé	Utilisateur du service TS	Propriétaire de la connexion TC
Demande de participation tardive	JR	Unidiffusé	Utilisateur du service TS	Propriétaire de la connexion TC
Confirmation de participation tardive	JC	Unidiffusé	Propriétaire de la connexion TC	Utilisateur du service TS
Demande de sortie d'un utilisateur	LR	Unidiffusé	Propriétaire de la connexion TC Utilisateur du service TS	Utilisateur du service TS Propriétaire de la connexion TC
Demande de fin de connexion	CT	Multidiffusé	Propriétaire de la connexion TC	Utilisateurs du service TS
Demande d'obtention de jeton	TGR	Unidiffusé	Utilisateur expéditeur SU Propriétaire de la connexion TC	Propriétaire de la connexion TC Utilisateur expéditeur SU

Tableau 2 – Paquets du protocole ECTP-5

Nom entier	Abréviation	Transport	De	A
Confirmation d'obtention de jeton	TGC	Unidiffusé	Propriétaire de la connexion TC Utilisateur expéditeur SU	Utilisateur expéditeur SU Propriétaire de la connexion TC
Demande de restitution de jeton	TRR	Unidiffusé	Utilisateur du service TS Propriétaire de la connexion TC	Propriétaire de la connexion TC Utilisateur du service TS
Confirmation de restitution de jeton	TRC	Unidiffusé	Propriétaire de la connexion TC Utilisateur du service TS	Utilisateur du service TS Propriétaire de la connexion TC
Rapport sur l'état des jetons	TSR	Multidiffusé Unidiffusé	Propriétaire de la connexion TC Propriétaire de la connexion TC	Utilisateurs du service TS Utilisateur du service TS
Demande de rapport sur l'état des jetons	TSRR	Unidiffusé	Utilisateur du service TS	Propriétaire de la connexion TC
Demande de changement d'arborescence	TCR	Unidiffusé	Propriétaire local/Entité feuille	Entité feuille
Confirmation de changement d'arborescence	TCC	Unidiffusé	Entité feuille	Propriétaire local/ Entité feuille
Demande de délégation d'arborescence	TDR	Unidiffusé	Propriétaire local/Entité feuille	Propriétaire local/ Entité feuille
Confirmation de délégation d'arborescence	TDC	Unidiffusé	Propriétaire local/Entité feuille	Propriétaire local/ Entité feuille
Demande de notification de changement d'arborescence	TNR	Unidiffusé	Entité feuille	Propriétaire local
Confirmation de notification de changement d'arborescence	TNC	Unidiffusé	Propriétaire local	Entité feuille
Demande de changement d'arborescence de gestion	CCR	Unidiffusé	Propriétaire local	Entité feuille
Confirmation de changement d'arborescence de gestion	CCC	Unidiffusé	Entité feuille	Propriétaire local

On trouvera dans le Tableau 3 la valeur de codage et la structure de chaque paquet ECTP-5. Les éléments d'extension sont joints à l'en-tête de base dans l'ordre défini dans le tableau.

Tableau 3 – Format des paquets ECTP-5

Type de paquet	Valeur de codage	Éléments d'extension ou données utilisateur (structure de paquet)	Longueur (octets)	Phase opérationnelle du protocole
CR	0000 0001	Connexion	20	Création de connexion
CC	0000 0010		16	
TJ	0000 0011	Horodateur	28	Gestion d'arborescence logique
TC	0000 0100	Horodateur	28	
TLR	0010 0011		16	
TLC	0010 0100		16	
DT	0000 0101	Données utilisateur	16+	Transport de données

Tableau 3 – Format des paquets ECTP-5

Type de paquet	Valeur de codage	Éléments d'extension ou données utilisateur (structure de paquet)	Longueur (octets)	Phase opérationnelle du protocole
RD	0000 0111	Horodateur + <i>Données utilisateur</i>	28+	Protection contre les erreurs
ACK	0000 1000	<i>Suite binaire d'erreurs</i>	16+	
NACK	0001 1000	Accusé de réception négatif + Horodateur	36	
PB	0000 1001		16	Gestion des membres
PBACK	0000 1110		16	
JR	0000 1010		16	Participation tardive
JC	0000 1011	Connexion	20	
LR	0000 1100		16	Sortie d'un utilisateur
CT	0000 1101		16	Fin
TGR	0001 0001	<i>Informations sur le propriétaire local</i>	16+	Obtention de jeton
TGC	0001 0010	Informations sur le propriétaire local	16+	
TRR	0001 0011		16	Restitution de jeton
TRC	0001 0100		16	
TSR	0001 0101	<i>Jeton + Informations sur le propriétaire local</i>	16+	Rapport sur les jetons
TSRR	0010 0101		16	
TCR	0001 0110	Informations de changement d'arborescence	24	Adaptation d'arborescence logique
TCC	0001 0111		16	
TDR	0001 1110	Informations de changement d'arborescence + Suite binaire d'erreurs	24+	
TDC	0001 1111		16	
TNR	0010 0001	Informations de changement d'arborescence	24	Gestion d'arborescence de gestion
TNC	0010 0010		16	
CCR	0010 1000	Informations de changement d'arborescence	24	
CCC	0010 1001		16	

Il convient de noter que les parties en italique (des éléments d'extension) indiquent que l'utilisation de l'élément d'extension correspondant est facultative et non impérative lors de l'implémentation. Dans la colonne relative à la longueur de paquet, le signe "+" signifie qu'on peut accroître la taille de paquet en ajoutant l'élément facultatif défini ou des données utilisateur.

En revanche, les valeurs de codages suivantes sont réservées pour une future utilisation: '0000 0000', '0000 1111', '0001 0000', '0010 0000' et '0000 0110'.

### 8.3.1 Demande de création de connexion (CR, *connection creation request*)

Le paquet CR est utilisé par le propriétaire de la connexion TC (TCN) pour créer une connexion multidiffusée n-plex. Le propriétaire de la connexion TC envoie le paquet CR au groupe aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC;
- port source: numéro de port local du propriétaire de la connexion TC;
- adresse IP de destination: adresse IP multidiffusion de groupe;
- port de destination: numéro de port de groupe.

La longueur du paquet CR est de 20 octets (en-tête de base de 16 octets + Élément Connexion de 4 octets).

Le paquet CR a le format suivant, comme le montre la Figure 21:

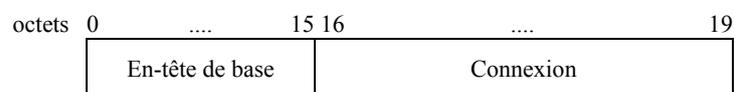


Figure 21 – Paquet CR

L'en-tête de base de 16 octets du paquet CR doit être codé comme suit:

*Élément suivant:* '0001' (élément Connexion).

*Version:* '00' (version du protocole ECTP-5).

*CT:* '11' (connexion multidiffusée n-plex).

*Type de paquet:* '0000 0001' (CR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local du propriétaire de la connexion TC (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*Longueur de capacité utile:* '4'.

*F:* '0' (à ne pas prendre en compte).

*ID de jeton:* '0' (l'ID de jeton du propriétaire de la connexion TC est mis à '0').

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Connexion de 4 octets doit être codé comme suit:

*Élément suivant:* '0000'.

*TCO:* tel que configuré par le propriétaire de la connexion TC.

*AGN:* tel que configuré par le propriétaire de la connexion TC.

*MSS:* tel que configuré par le propriétaire de la connexion TC (la valeur par défaut étant '1024').

### 8.3.2 Confirmation de création de connexion (CC, *connection creation confirm*)

Le paquet CC est utilisé par l'utilisateur du service TS en réponse au paquet CR du propriétaire de la connexion TC (TCN). Un utilisateur du service TS envoie le paquet CC au propriétaire de la connexion aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'utilisateur du service TS;
- port source: numéro de port local de l'utilisateur du service TS;
- adresse IP de destination: adresse IP du propriétaire de la connexion TC;
- port de destination: numéro de port de groupe.

Le paquet CC contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0000 0010' (CC).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'utilisateur du service TS (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

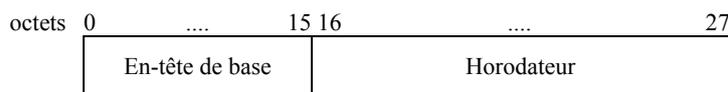
Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.3 Demande de participation à une arborescence (TJ, *tree join request*)

Le paquet TJ est envoyé par une entité feuille (LE) ou un propriétaire local (LO) au propriétaire local ou à l'entité feuille concernés en vue d'une participation à une arborescence intra-groupe ou inter-groupes. L'adresse utilisée est la suivante:

- adresse IP source: adresse IP de l'entité feuille ou du propriétaire local;
- port source: numéro de port local de l'entité feuille ou du propriétaire local;
- adresse IP de destination: adresse IP du propriétaire local ou de l'entité feuille;
- port de destination: numéro de port de groupe.

Le paquet TJ contient l'en-tête de base de 16 octets et l'élément Horodateur de 12 octets. Son format est le suivant, comme le montre la Figure 22:



**Figure 22 – Paquet TJ**

L'en-tête de base du paquet TJ doit être codé comme suit:

*Élément suivant:* '0100' (élément Horodateur).

*CT:* '11'.

*Type de paquet:* '0000 0011' (TJ).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro de séquence du paquet.

*Longueur de capacité utile:* '12'.

*F:* '1' (pour la participation à une arborescence inter-groupes), '0' (pour la participation à une arborescence intra-groupe).

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Horodateur de 12 octets doit être codé comme suit:

*Élément suivant:* '0000'.

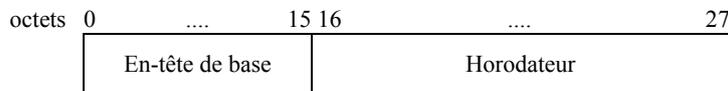
*Horodateur:* heure qu'il est au moment de l'envoi du paquet.

### 8.3.4 Confirmation de participation à une arborescence (TC, *tree join confirm*)

Le paquet TC est envoyé par le nœud propriétaire local ou entité feuille, en réponse au paquet TJ. Le propriétaire local ou l'entité feuille envoie le paquet TC à la partie qui demande à se joindre à une arborescence (entité feuille ou propriétaire local) aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire local ou de l'entité feuille;
- port source: numéro de port de groupe;
- adresse IP de destination: adresse IP de l'entité feuille ou du propriétaire local;
- port de destination: port local de l'entité feuille ou du propriétaire local.

Le paquet TC contient l'en-tête de base de 16 octets et l'élément Horodateur de 12 octets, comme l'indique la Figure 23.



**Figure 23 – Paquet TC**

L'en-tête de base de 16 octets du paquet TC doit être codé comme suit:

*Élément suivant:* '0100' (élément Horodateur).

*CT:* '11'.

*Type de paquet:* '0000 0100' (TC).

*Total de contrôle:* à calculer.

*Port source:* port de groupe (ou ID de connexion).

*Port de destination:* port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*PSN:* valeur reprise du champ *PSN* du paquet de demande correspondant.

*Longueur de capacité utile:* '12'.

*F:* '1' si la demande TJ est acceptée, '0' dans le cas contraire.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Horodateur de 12 octets doit être codé comme suit:

*Élément suivant:* '0000'.

*Horodateur:* valeur temporelle de 8 octets reprise du paquet TJ correspondant.

### 8.3.5 Demande de sortie d'arborescence (TLR, *tree leave request*)

Le paquet TLR est envoyé par l'entité feuille (LE) ou le propriétaire local (LO) au propriétaire local ou à l'entité feuille qui est le parent en vue d'une sortie de l'arborescence logique. Le paquet TLR est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille ou du propriétaire local;
- port source: port local de l'entité feuille ou du propriétaire local;
- adresse IP de destination: adresse IP du propriétaire local ou de l'entité feuille;
- port de destination: numéro de port de groupe.

Le paquet TLR contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0010 0011' (TLR).

*Total de contrôle:* à calculer.

*Port source:* port local de l'entité feuille ou du propriétaire local (ou ID de connexion)

*Port de destination:* port de groupe (ou ID de connexion).

*PSN:* numéro de séquence du paquet.

*F:* '1' (pour la participation à une arborescence inter-groupes), '0' (pour la participation à une arborescence intra-groupe).

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.6 Confirmation de sortie d'arborescence (TLC, *tree leave confirm*)

Le paquet TLC est envoyé par le nœud propriétaire local ou entité feuille en réponse au paquet TLR. Le propriétaire local ou l'entité feuille envoie le paquet TLC à l'entité feuille ou au propriétaire local enfant aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire local ou de l'entité feuille;
- port source: numéro de port de groupe;
- adresse IP de destination: adresse IP de l'entité feuille ou du propriétaire local;
- port de destination: port local de l'entité feuille ou du propriétaire local.

Le paquet TLC contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0010 0100' (TLC).

*Total de contrôle:* à calculer.

*Port source:* numéro de port de groupe (ou ID de connexion).

*Port de destination:* numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*PSN:* valeur reprise du champ PSN du paquet de demande correspondant.

*F:* '1' si la demande TLC est acceptée, '0' dans le cas contraire.

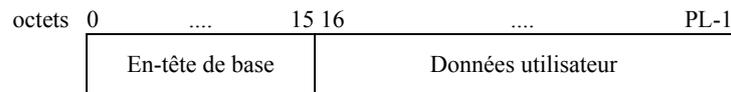
Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne devront pas être pris en compte par le destinataire.

### 8.3.7 Données (DT, *data*)

Le paquet DT est utilisé par le propriétaire de la connexion ou l'utilisateur expéditeur SU pour transmettre les données multidiffusées aux membres du groupe. Il est également utilisé par le propriétaire local pour transmettre le trafic de test en vue d'une adaptation d'arborescence logique. Les paquets DT multidiffusés sont envoyés aux utilisateurs du service TS aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC, de l'utilisateur expéditeur SU ou du propriétaire local;
- port source: numéro de port local du propriétaire de la connexion TC, de l'utilisateur expéditeur SU ou du propriétaire local;
- adresse IP de destination: adresse IP multidiffusion de groupe;
- port de destination: numéro de port de groupe.

Le paquet DT contient l'en-tête de base de 16 octets et les données utilisateur de longueur variable comme le montre la Figure 24:



**Figure 24 – Paquet DT**

L'en-tête de base de 16 octets du paquet DT doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0000 0101' (DT).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local du propriétaire de la connexion TC, de l'utilisateur expéditeur ou du propriétaire local (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro *PSN* du paquet DT, qui commence par le numéro de séquence initial, augmente de '1' et reboucle sur '1' après avoir atteint la valeur  $2^{32} - 1$ .

*Longueur de capacité utile:* indique la longueur (en octets) des données utilisateur contenues dans le paquet.

*F:* '1' (pour le trafic de test du propriétaire local), sinon '0'.

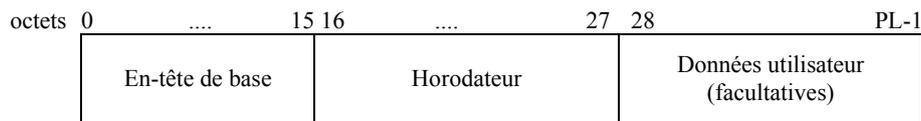
*ID de jeton:* ID de jeton de l'expéditeur du paquet de données ('0' pour le propriétaire de la connexion TC ou un nombre positif pour l'utilisateur expéditeur SU).

### 8.3.8 Données de retransmission (RD, *retransmission data*)

Le paquet RD est envoyé par le propriétaire local ou l'entité feuille pour retransmettre des données en réponse à des demandes de réparation (NACK) émanant de nœuds enfants (entité feuille ou propriétaire local) le long de l'arborescence de gestion. Le format de paquet est le même que celui du paquet DT. Cependant, au contraire de la transmission de paquets DT multidiffusés, le paquet RD est envoyé au demandeur en mode unidiffusion aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire local ou de l'entité feuille;
- port source: numéro de port local du propriétaire local ou de l'entité feuille;
- adresse IP de destination: adresse IP du demandeur de la réparation (entité feuille ou propriétaire local);
- port de destination: numéro de port de groupe.

Le paquet RD contient l'en-tête de base de 16 octets et les données utilisateur de longueur variable, comme le montre la Figure 25:



**Figure 25 – Paquet RD**

L'en-tête de base de 16 octets du paquet RD doit être codé comme suit:

*Élément suivant:* '0100' (élément Horodateur).

*CT:* '11'.

*Type de paquet:* '0000 0111' (RD).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local du propriétaire local ou de l'entité feuille (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro *PSN* du paquet de données dont la retransmission est demandée.

*Longueur de capacité utile:* indique la longueur (en octets) des données utilisateur contenues dans le paquet.

*F:* '1' si l'expéditeur du paquet n'a pas de données à réparer, '0' dans le cas contraire.

*ID de jeton:* ID de jeton de l'expéditeur du paquet de données ('0' pour le propriétaire de la connexion TC ou un nombre positif pour l'utilisateur expéditeur SU).

L'élément Horodateur de 12 octets doit être codé comme suit:

*Élément suivant:* '0000'.

*Horodateur:* valeur temporelle de 8 octets reprise du paquet NACK correspondant.

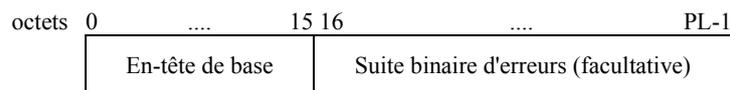
### 8.3.9 Accusé de réception (ACK, *acknowledgement*)

Le paquet ACK est envoyé au nœud parent le long de l'arborescence de gestion pour accuser réception des paquets DT reçus d'un utilisateur expéditeur SU. Il est créé selon la règle de création des paquets ACK, qui sera décrite ultérieurement. Si le paquet ACK est destiné au trafic de test du propriétaire local ainsi qu'il est décrit au § 7.5, il devrait contenir un élément Suite binaire d'erreurs en vue de l'adaptation d'arborescence logique.

Sur la voie de gestion unidiffusion, un paquet ACK est envoyé par une entité feuille ou un propriétaire local à son parent (entité feuille ou propriétaire local) le long de l'arborescence de gestion dont la racine est l'utilisateur expéditeur SU des données correspondantes. Il est transmis aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille ou du propriétaire local;
- port source: numéro de port local de l'entité feuille ou du propriétaire local;
- adresse IP de destination: adresse IP du nœud parent (propriétaire local ou entité feuille);
- port de destination: numéro de port de groupe.

Le paquet ACK contient l'en-tête de base de 16 octets et, facultativement, un élément Suite binaire d'erreurs de longueur variable comme indiqué dans la Figure 26:



**Figure 26 – Paquet ACK**

L'en-tête de base du paquet ACK doit être codé comme suit:

*Élément suivant:* '0000' ou '0010' (élément Suite binaire d'erreurs) pour le paquet ACK en réponse au trafic de test en provenance du propriétaire local.

*CT:* '11'.

*Type de paquet:* '0000 1000' (ACK).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* LSN, numéro PSN du paquet DT au numéro est le plus bas qui n'a pas encore été reçu.

*Longueur de capacité utile:* longueur (en octets) des éléments d'extension joints à l'en-tête de base.

*F:* '0' (à ne pas prendre en compte).

*ID de jeton:* ID de jeton de l'expéditeur correspondant ('0' pour le propriétaire de la connexion TC ou un nombre positif pour l'utilisateur expéditeur SU).

L'élément Suite binaire d'erreurs doit être codé comme suit:

*Élément suivant:* '0000'.

*Longueur de suite binaire:* représente la longueur totale de la *suite binaire d'erreurs* en mots (4 octets).

*Longueur de suite binaire valide:* longueur réellement valide de la *suite binaire d'erreurs* en bits.

*Suite binaire d'erreurs:* représente les informations de suite binaire indiquant quels paquets DT sont perdus.

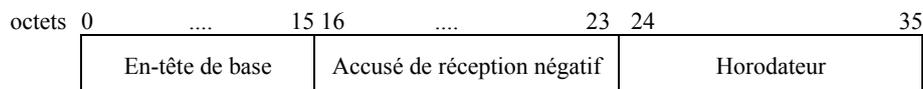
### 8.3.10 Accusé de réception négatif (NACK, *negative acknowledgement*)

Le paquet NACK permet de demander à un parent de retransmettre des données dont la perte a été détectée. Un paquet NACK est envoyé immédiatement à la détection d'une perte de paquets. Au moyen des champs *Nombre de paquets perdus* et *Numéro de séquence du paquet de départ* de l'élément Accusé de réception négatif, un paquet NACK peut signaler une demande de réparation pour un bloc de deux ou de plusieurs paquets consécutifs perdus.

Sur la voie de gestion unidiffusion, un paquet NACK est envoyé par une entité feuille ou un propriétaire local à son nœud parent le long de l'arborescence de gestion dont la racine est l'utilisateur expéditeur SU des données correspondantes. Il est transmis aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille ou du propriétaire local;
- port source: numéro de port local de l'entité feuille ou du propriétaire local;
- adresse IP de destination: adresse IP du nœud parent (propriétaire local ou entité feuille);
- port de destination: numéro de port de groupe.

Le paquet NACK contient l'en-tête de base de 16 octets, l'élément Accusé de réception négatif de 8 octets et l'élément Horodateur de 12 octets, comme indiqué dans la Figure 27.



**Figure 27 – Paquet NACK**

L'en-tête de base du paquet NACK doit être codé comme suit:

*Élément suivant:* '1000' (accusé de réception négatif).

*CT:* '11'.

*Type de paquet:* '0001 1000' (NACK).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro LSN qui est le numéro PSN du paquet DT au numéro est le plus bas qui n'a pas encore été reçu.

*Longueur de capacité utile:* '20'.

*F*: '0' (à ne pas prendre en compte).

*ID de jeton*: ID de jeton de l'expéditeur correspondant.

L'élément Accusé de réception négatif doit être codé comme suit:

*Élément suivant*: '0100' (élément Horodateur).

*Nombre de paquets perdus*: représente le nombre de paquets consécutifs perdus à partir du *Numéro de séquence du paquet de départ*.

*Numéro de séquence du paquet de départ*: Numéro de séquence du paquet de départ d'un bloc de paquets perdus.

L'élément Horodateur de 12 octets doit être codé comme suit:

*Élément suivant*: '0000'.

*Horodateur*: heure qu'il est au moment de l'envoi du paquet.

### 8.3.11 Sonde (PB, *probe*)

Le paquet PB est utilisé par le propriétaire de la connexion TC (TCN) pour la mise à jour de la connexion. Pendant la session, le propriétaire de la connexion TC envoie périodiquement un paquet PB à un des utilisateurs du service TS aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC;
- port source: numéro de port local du propriétaire de la connexion TC;
- adresse IP de destination: adresse IP de l'utilisateur du service TS choisi;
- port de destination: numéro de port de groupe.

Le paquet PB contient uniquement l'en-tête de base de 16 octets, dont le format est le suivant:

*Élément suivant*: '0000'.

*CT*: '11'.

*Type de paquet*: '0000 1001' (PB).

*Total de contrôle*: à calculer.

*Port source*: numéro de port local du propriétaire de la connexion TC (ou ID de connexion).

*Port de destination*: numéro de port de groupe (ou ID de connexion).

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.12 Accusé de réception de sonde (PBACK, *probe acknowledgement*)

Le paquet PBACK est envoyé par un utilisateur du service TS en réponse au paquet PB envoyé par le propriétaire de la connexion TC. Lorsqu'un utilisateur du service TS reçoit le paquet PB du propriétaire de la connexion TC, il devrait répondre par un paquet PBACK, qui indique qu'il est toujours actif.

L'utilisateur du service TS envoie le paquet PBACK au propriétaire de la connexion aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'utilisateur du service TS;
- port source: numéro de port local de l'utilisateur du service TS;
- adresse IP de destination: adresse IP du propriétaire de la connexion TC;
- port de destination: numéro de port local du propriétaire de la connexion TC.

Le paquet PBACK contient uniquement l'en-tête de base de 16 octets, dont le format est le suivant:

*Élément suivant*: '0000'.

*CT*: '11'.

*Type de paquet*: '0000 1110' (PBACK).

*Total de contrôle*: à calculer.

*Port source*: numéro de port local de l'utilisateur du service TS (ou ID de connexion).

*Port de destination*: numéro de port local du propriétaire de la connexion TC (ou ID de connexion).

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

**8.3.13 Demande de participation tardive (JR, *late join request*)**

Le paquet JR est utilisé par un nouvel utilisateur du service TS participant pour se joindre à la connexion ECTP. Ce dernier envoie le paquet JR au propriétaire de la connexion TC aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'utilisateur du service TS;
- port source: numéro de port local de l'utilisateur du service TS;
- adresse IP de destination: adresse IP du propriétaire de la connexion TC;
- port de destination: numéro de port de groupe.

Le paquet JR contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0000 1010' (JR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'utilisateur du service TS (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro de séquence de paquet.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

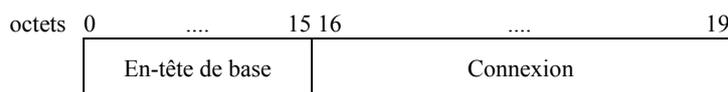
**8.3.14 Confirmation de participation tardive (JC, *late join confirm*)**

Le paquet JC est utilisé par le propriétaire de la connexion TC, en réponse au paquet JR. Le propriétaire de la connexion TC envoie le paquet JC au nouvel utilisateur du service TS participant aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC;
- port source: numéro de port local du propriétaire de la connexion TC;
- adresse IP de destination: adresse IP de l'utilisateur du service TS;
- port de destination: port local de l'utilisateur du service TS.

Le paquet JC contient l'en-tête de base de 16 octets et l'élément Connexion de 4 octets.

Il a le format suivant, comme le montre la Figure 28:



**Figure 28 – Paquet JC**

L'en-tête de base de 16 octets du paquet JC doit être codé comme suit:

*Élément suivant:* '0001' (élément Connexion).

*CT:* '11'.

*Type de paquet:* '0000 1011' (JC).

*Total de contrôle:* à calculer.

*Port source:* port local du propriétaire de la connexion TC (ou ID de connexion).

*Port de destination:* port local de l'utilisateur du service TS (ou ID de connexion).

*PSN:* valeur reprise du champ PSN du paquet de demande correspondant.

*Longueur de capacité utile:* '4'.

*F:* '1' si la demande JR est acceptée, '0' dans le cas contraire.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Connexion de 4 octets doit être codé comme suit:

*Élément suivant:* '0000'.

*TCO:* tel que configuré par le propriétaire de la connexion TC.

*AGN:* tel que configuré par le propriétaire de la connexion TC.

*MSS:* tel que configuré par le propriétaire de la connexion TC.

### 8.3.15 Demande de sortie d'un utilisateur (LR, *user leave request*)

Le paquet LR est utilisé par l'utilisateur du service TS pour indiquer qu'il quittera la connexion, ou par le propriétaire de la connexion TC pour éjecter un utilisateur du service TS perturbateur. Le paquet LR n'exige pas de paquet de confirmation correspondant. Il est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'utilisateur du service TS (sortie d'un utilisateur) ou du propriétaire de la connexion TC (éjection des éléments perturbateurs);
- port source: numéro de port local de l'utilisateur du service TS ou du propriétaire de la connexion TC;
- adresse IP de destination: adresse IP du propriétaire de la connexion TC ou de l'utilisateur du service TS;
- port de destination: numéro de port de groupe ou numéro de port local de l'utilisateur du service TS.

Le paquet LR contient uniquement l'en-tête de base de 16 octets, qui a le format suivant:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0000 1100' (LR).

*Total de contrôle:* à calculer.

*Port source:* port local de l'utilisateur du service TS ou du propriétaire de la connexion TC (ou ID de connexion).

*Port de destination:* port de groupe ou port local de l'utilisateur du service TS (ou ID de connexion).

*F:* '1' pour une sortie demandée par l'utilisateur, ou '0' pour l'éjection des éléments perturbateurs.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.16 Demande de fin de connexion (CT, *connection termination request*)

Le paquet CT est utilisé par le propriétaire de la connexion TC (TCN) pour mettre fin à la connexion. Il n'exige pas de paquet de confirmation correspondant. Il est envoyé par le propriétaire de la connexion TC aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC;
- port source: numéro de port local du propriétaire de la connexion TC;
- adresse IP de destination: adresse IP multidiffusion de groupe;
- port de destination: numéro de port de groupe.

Le paquet CT contient uniquement l'en-tête de base de 16 octets, qui a le format suivant:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0000 1101' (CT).

*Total de contrôle:* à calculer.

*Port source:* port local du propriétaire de la connexion TC (ou ID de connexion).

*Port de destination:* port de groupe (ou ID de connexion).

*F:* '1' pour une fin anormale, ou '0' pour une fin normale (après la transmission des données multidiffusées).

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.17 Demande d'obtention de jeton (TGR, *token get request*)

Le paquet TGR est utilisé par un utilisateur du service TS pour obtenir un jeton en vue du transport de données multidiffusées (opération d'obtention de jeton initiée par l'utilisateur du service TS). Dans ce cas, l'utilisateur du service

## ISO/CEI 14476-5:2008 (F)

TS peut demander un jeton au propriétaire de la connexion TC en envoyant un paquet TGR. L'utilisateur du service TS qui détient un jeton devient un utilisateur expéditeur SU.

Dans l'opération de cession de jeton initiée par le propriétaire de la connexion TC, ce dernier demande à un utilisateur du service TS de devenir un utilisateur expéditeur SU. Dans ce cas, le propriétaire de la connexion TC enverra un paquet TGR à un utilisateur du service TS déterminé.

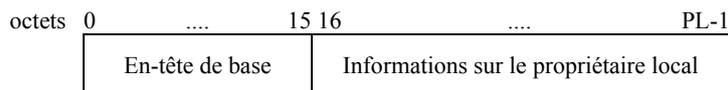
Dans le cas de l'opération d'obtention de jeton, l'élément Informations sur le propriétaire local devrait être joint au paquet TGR par l'utilisateur du service TS. Il donne au propriétaire de la connexion TC des renseignements sur le propriétaire local auquel se joint l'utilisateur du service TS. Ces informations sont utilisées lors de l'envoi de paquets TSR.

Le paquet TGR est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'utilisateur du service TS (obtention de jeton) ou adresse IP du propriétaire de la connexion TC (cession de jeton);
- port source: numéro de port local de l'utilisateur du service TS (obtention de jeton) ou numéro de port local du propriétaire de la connexion TC (cession de jeton);
- adresse IP de destination: adresse IP du propriétaire de la connexion TC (obtention de jeton) ou adresse IP de l'utilisateur du service TS (cession de jeton);
- port de destination: numéro de port de groupe (obtention de jeton) ou numéro de port local de l'utilisateur du service TS (cession de jeton).

Le paquet TGR contient l'en-tête de base de 16 octets et l'élément Informations sur le propriétaire local de longueur variable.

Il a le format suivant, comme le montre la Figure 29:



**Figure 29 – Paquet TGR**

L'en-tête de base de 16 octets du paquet TGR doit être codé comme suit:

*Élément suivant:* '0000' ou '0111' (Élément Informations sur le propriétaire local, dans le cas d'une obtention de jeton).

*CT:* '11'.

*Type de paquet:* '0001 0001' (TGR).

*Total de contrôle:* à calculer.

*port source:* numéro de port local de l'utilisateur du service TS ou du propriétaire de la connexion TC (ou ID de connexion).

*Port de destination:* numéro de port de groupe ou port local de l'utilisateur du service TS (ou ID de connexion).

*PSN:* numéro de séquence de paquet.

*Longueur de capacité utile:* longueur (en octets) des éléments d'extension joints à l'en-tête de base.

*F:* à ne pas prendre en compte.

*ID de jeton:* ID de jeton attribué par le propriétaire de la connexion TC (cession de jeton); dans le cas de l'obtention de jeton, ce champ n'est pas pris en compte.

Si l'élément Informations sur le propriétaire local est ajouté à l'élément suivant dans le cas d'une obtention de jeton, il doit être codé comme suit:

*Élément suivant:* '0000'.

*Nombre de jetons:* '1'.

*ID du propriétaire local:* représente l'ID du propriétaire local auquel l'entité feuille se joint.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.18 Confirmation d'obtention de jeton (TGC, *token get confirm*)

En réponse à un paquet TGR, le propriétaire de la connexion TC envoie un paquet TGC à l'utilisateur du service TS qui a envoyé le paquet TGR (opération d'obtention de jeton initiée par l'utilisateur du service TS). Dans l'opération de cession de jeton initiée par le propriétaire de la connexion TC, le paquet TGC est utilisé par l'utilisateur du service TS pour confirmer la demande.

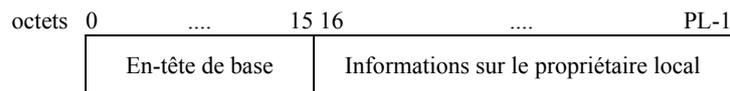
Dans le cas d'une cession de jeton, l'élément Informations sur le propriétaire local devrait être ajouté au paquet TGC par l'utilisateur du service TS. Il donne au propriétaire de la connexion TC des informations sur le propriétaire local auquel se joint l'entité feuille. Ces informations sont utilisées lors de l'envoi de paquets TSR.

Le paquet TGC est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC (obtention de jeton) ou adresse IP de l'entité feuille (cession de jeton);
- port source: numéro de port de groupe (obtention de jeton) ou numéro de port local de l'entité feuille (cession de jeton);
- adresse IP de destination: adresse IP de l'entité feuille (obtention de jeton) ou adresse IP du propriétaire de la connexion TC (cession de jeton);
- port de destination: port local de l'entité feuille (obtention de jeton) ou port local du propriétaire de la connexion TC (cession de jeton).

Le paquet TGC contient l'en-tête de base de 16 octets et l'élément Informations sur le propriétaire local de longueur variable.

Le paquet TGC a le format suivant, comme le montre la Figure 30:



**Figure 30 – Paquet TGC**

L'en-tête de base de 16 octets du paquet TGC doit être codé comme suit:

*Élément suivant:* '0000' ou '0111' (Élément Informations sur le propriétaire local dans le cas d'une cession de jeton).

*CT:* '11'.

*Type de paquet:* '0001 0010' (TGC).

*Total de contrôle:* à calculer.

*Port source:* port de groupe ou numéro de port local de l'entité feuille (ou ID de connexion).

*Port de destination:* numéro de port de groupe ou port local de l'entité feuille (ou ID de connexion).

*PSN:* valeur reprise du champ PSN du paquet de demande correspondant.

*Longueur de capacité utile:* longueur (en octets) des éléments d'extension joints à l'en-tête de base.

*F:* '0' (acceptation) ou '1' (rejet).

*ID de jeton:* ID de jeton attribué par le propriétaire de la connexion TC (obtention de jeton); dans le cas d'une cession de jeton, ce champ n'est pas pris en compte.

Si l'élément Informations sur le propriétaire local est ajouté en tant qu'élément suivant dans le cas d'une cession de jeton, il doit être codé comme suit:

*Élément suivant:* '0000'.

*Nombre de jetons:* '1'.

*ID du propriétaire local:* représente l'ID du propriétaire local auquel l'entité feuille se joint.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.19 Demande de restitution de jeton (TRR, *token return request*)

Le paquet TRR est utilisé par un utilisateur du service TS pour restituer un jeton au propriétaire de la connexion TC (opération de restitution de jeton initiée par un utilisateur du service TS). Dans ce cas, l'utilisateur du service TS envoie un paquet TRR. Dans le cas d'un retrait de jeton activé par le propriétaire de la connexion TC, celui-ci peut d'abord demander à un utilisateur du service TS de restituer le jeton. Le propriétaire de la connexion TC enverra alors un paquet TRR à l'utilisateur du service TS concerné.

Le paquet TGR est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'utilisateur du service TS (restitution de jeton) ou adresse IP du propriétaire de la connexion TC (retrait de jeton);
- port source: port local de l'utilisateur du service TS (restitution de jeton) ou port local du propriétaire de la connexion TC (retrait de jeton);
- adresse IP de destination: adresse IP du propriétaire de la connexion TC (restitution de jeton) ou adresse IP de l'utilisateur du service TS (retrait de jeton);
- port de destination: port de groupe (restitution de jeton) ou port local de l'utilisateur du service TS (retrait de jeton).

Le paquet TRR contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0001 0011' (TRR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'utilisateur du service TS ou du propriétaire de la connexion TC (ou ID de connexion).

*Port de destination:* numéro de port de groupe ou port local de l'utilisateur du service TS (ou ID de connexion).

*PSN:* numéro de séquence de paquet.

*ID de jeton:* ID de jeton de l'utilisateur du service TS.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.20 Confirmation de restitution de jeton (TRC, *token return confirm*)

Le paquet TRC est utilisé par le propriétaire de la connexion TC ou l'utilisateur du service TS pour confirmer la demande TRR connexe. Il est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC (restitution de jeton) ou adresse IP de l'utilisateur du service TS (retrait de jeton);
- port source: port de groupe (restitution de jeton) ou port local de l'utilisateur du service TS (retrait de jeton);
- adresse IP de destination: adresse IP de l'utilisateur du service TS (restitution de jeton) ou adresse IP du propriétaire de la connexion TC (retrait de jeton);
- port de destination: port local de l'utilisateur du service TS (restitution de jeton) ou port local du propriétaire de la connexion TC (retrait de jeton).

Le paquet TRC contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0001 0100' (TRC).

*Total de contrôle:* à calculer.

*Port source:* port de groupe ou numéro de port local de l'utilisateur du service TS (ou ID de connexion).

*Port de destination:* numéro de port de groupe ou port local de l'utilisateur du service TS (ou ID de connexion).

*PSN:* valeur reprise du champ PSN du paquet de demande correspondant.

*ID de jeton:* ID de jeton de l'utilisateur du service TS.

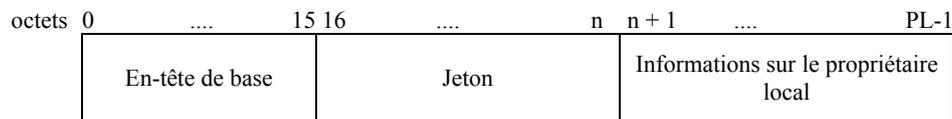
Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.21 Rapport sur l'état des jetons (TSR, *token status report*)

Le paquet TSR est utilisé par le propriétaire de la connexion TC pour annoncer l'ID de jeton valide au même moment dans la connexion. Il donne aussi aux propriétaires locaux des renseignements sur les ID de jetons correspondants de chaque propriétaire local en joignant l'élément Informations sur le propriétaire local. Le paquet TSR est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire de la connexion TC;
- port source: port local du propriétaire de la connexion TC ou port de groupe (en réponse au paquet TSRR);
- adresse IP de destination: adresse IP multidiffusion de groupe ou adresse IP de l'utilisateur du service TS (en réponse au paquet TSRR);
- port de destination: port de groupe ou port local de l'utilisateur du service TS (en réponse au paquet TSRR).

Le paquet TSR a le format suivant, comme le montre la Figure 31:



**Figure 31 – Paquet TSR**

Le paquet TSR contient l'en-tête de base de 16 octets, l'élément Jeton de longueur variable et l'élément Informations sur le propriétaire local de longueur variable. L'en-tête de base doit être codé comme suit:

*Élément suivant:* '0110' (élément Jeton).

*CT:* '11'.

*Type de paquet:* '0001 0101' (TSR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local du propriétaire de la connexion TC (ou ID de connexion) ou port de groupe (en réponse au paquet TSRR).

*Port de destination:* numéro de port de groupe (ou ID de connexion) ou port local de l'utilisateur du service TS (en réponse au paquet TSRR).

*Longueur de capacité utile:* longueur (en octets) des éléments d'extension joints à l'en-tête de base.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Jeton de longueur variable doit être codé comme suit:

*Élément suivant:* '0111' (élément Informations sur le propriétaire local).

*Nombre de jetons:* indique le nombre total d'*ID de jetons valides* dans la connexion.

*ID de jetons valides:* contient la liste des ID de jetons valides dans la connexion. La longueur de chaque *ID de jeton* est de 1 octet.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Informations sur le propriétaire local de longueur variable doit être codé comme suit:

*Élément suivant:* '0000' ou '0111' (si un autre élément Informations sur le propriétaire local suit).

*Nombre de jetons:* nombre total d'*ID de jetons correspondants* pour le propriétaire local dont l'*ID* est l'*ID du propriétaire local*.

*ID du propriétaire local:* représente l'*ID du propriétaire local* auquel se joint l'entité feuille.

*ID de jetons correspondants:* contient la liste des ID de jetons correspondant à l'*ID du propriétaire local* dans la connexion. La longueur de chaque *ID de jeton* est de 1 octet.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.22 Demande de rapport sur l'état des jetons (TSRR, *token status report request*)

Le paquet TSRR est utilisé par l'utilisateur du service TS pour demander au propriétaire de la connexion TC de mettre à jour la liste de jetons. Un utilisateur du service TS envoie le paquet TSRR au propriétaire de la connexion TC aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'utilisateur du service TS;
- port source: numéro de port local de l'utilisateur du service TS;
- adresse IP de destination: adresse IP du propriétaire de la connexion TC;
- Port de destination: numéro de port de groupe.

Le paquet TSRR contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0010 0101' (TSRR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'utilisateur du service TS (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

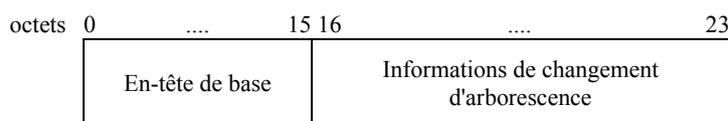
Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.23 Demande de changement d'arborescence (TCR, *tree change request*)

Le paquet TCR permet de changer la relation parent-enfant des nœuds de l'arborescence intra-groupe. Il ordonne au nœud qui le reçoit de devenir un enfant d'un autre nœud désigné par l'*ID de nœud*. Le paquet TCR est envoyé sur la voie de gestion unidiffusion aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille ou du propriétaire local;
- port source: numéro de port local de l'entité feuille ou du propriétaire local;
- adresse IP de destination: adresse IP du nœud cible (entité feuille);
- port de destination: numéro de port de groupe.

Le paquet TCR contient l'en-tête de base de 16 octets et l'élément Informations de changement d'arborescence de 8 octets comme indiqué dans la Figure 32.



**Figure 32 – Paquet TCR**

L'en-tête de base du paquet TCR doit être codé comme suit:

*Élément suivant:* '1001' (élément Informations de changement d'arborescence).

*CT:* '11'.

*Type de paquet:* '0001 0110' (TCR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro de séquence de paquet.

*Longueur de capacité utile:* '8'.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Informations de changement d'arborescence de 8 octets doit être codé comme suit:

*Élément suivant:* '0000'.

*ID de nœud:* représente un nouveau nœud parent du nœud qui reçoit le paquet.

### 8.3.24 Confirmation de changement d'arborescence (TCC, *tree change confirm*)

Le paquet TCC est envoyé par l'entité feuille (LE) en réponse à un paquet TCR. Le nœud envoie le paquet TCC à un nœud qui a envoyé le paquet TCR aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille;
- port source: numéro de port de groupe;
- adresse IP de destination: adresse IP de l'entité feuille ou du propriétaire local;
- port de destination: port local de l'entité feuille ou du propriétaire local.

Le paquet TCC contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0001 0111' (TCC).

*Total de contrôle:* à calculer.

*Port source:* numéro de port de groupe (ou ID de connexion).

*Port de destination:* numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*PSN:* valeur reprise du champ PSN du paquet de demande correspondant.

*F:* est mis à '1' si la demande TCC est acceptée, '0' dans le cas contraire.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

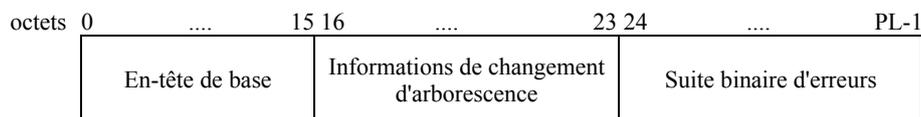
### 8.3.25 Demande de délégation d'arborescence (TDR, *tree delegation request*)

Le paquet TDR permet de déléguer le processus pour déterminer une position appropriée d'un des nœuds enfants par rapport à un autre nœud dans l'arborescence intra-groupe. Le paquet comprend l'*ID de nœud*. Le nœud recevant le paquet devrait trouver une meilleure relation parent-enfant pour un nœud ayant l'*ID de nœud* concerné au moyen des suites binaires d'erreurs disponibles.

Sur la voie de gestion unidiffusion, un TDR paquet est envoyé par tout nœud qui a des nœuds enfants à ses nœuds parents ou enfants dans la même arborescence intra-groupe. Il est transmis aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille ou du propriétaire local;
- port source: numéro de port local de l'entité feuille ou du propriétaire local;
- adresse IP de destination: adresse IP du nœud parent potentiel (propriétaire local ou entité feuille);
- port de destination: numéro de port de groupe.

Le paquet TDR contient l'en-tête de base de 16 octets, l'élément Informations de changement d'arborescence de 8 octets et un élément Suite binaire d'erreurs de longueur variable, comme indiqué dans la Figure 33.



**Figure 33 – Paquet TDR**

L'en-tête de base du paquet TDR doit être codé comme suit:

*Élément suivant:* '1001' (élément Informations de changement d'arborescence).

*CT:* '11'.

*Type de paquet:* '0001 1110' (TDR).

*Total de contrôle:* à calculer.

## ISO/CEI 14476-5:2008 (F)

*Port source*: numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*Port de destination*: numéro de port de groupe (ou ID de connexion).

*PSN*: numéro de séquence de paquet.

*Longueur de capacité utile*: longueur (en octets) des éléments d'extension joints à l'en-tête de base.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Informations de changement d'arborescence de 8 octets doit être codé comme suit:

*Élément suivant*: '0010' (élément Suite binaire d'erreurs).

*ID de nœud*: représente un enfant potentiel du nœud qui a reçu le paquet TDR.

L'élément Suite binaire d'erreurs doit être codé comme suit:

*Élément suivant*: '0000'.

*Longueur de suite binaire*: représente la longueur totale de la *suite binaire d'erreurs* en mots (4 octets).

*Longueur de suite binaire valide*: longueur réellement valide de la *suite binaire d'erreurs* en bits.

*Suite binaire d'erreurs*: représente les informations de suite binaire indiquant quels paquets DT sont perdus.

### 8.3.26 Confirmation de délégation d'arborescence (TDC, *tree delegation confirm*)

Le paquet TDC est envoyé en réponse à un paquet TDR. Le nœud envoie le paquet TDC à un nœud qui a envoyé le paquet TDR aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire local ou de l'entité feuille;
- port source: numéro de port de groupe;
- adresse IP de destination: adresse IP de l'entité feuille ou du propriétaire local;
- port de destination: port local de l'entité feuille ou du propriétaire local.

Le paquet TDC contient uniquement l'en-tête de base de 16 octets qui doit être codé comme suit:

L'en-tête de base du paquet TDC doit être codé comme suit:

*Élément suivant*: '0000'.

*CT*: '11'.

*Type de paquet*: '0001 1111' (TDC).

*Total de contrôle*: à calculer.

*Port source*: numéro de port de groupe (ou ID de connexion).

*Port de destination*: numéro de port local de l'entité feuille ou du propriétaire local (ou ID de connexion).

*PSN*: valeur reprise du champ PSN du paquet de demande correspondant.

*F*: est mis à '1' si la demande TDC est acceptée, '0' dans le cas contraire.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.27 Demande de notification de changement d'arborescence (TNR, *tree change notification request*)

Le paquet TNR permet de notifier le changement d'arborescence logique (arborescence intra-groupe) au propriétaire local (LO). À réception du paquet, le propriétaire local devrait indiquer à son arborescence intra-groupe le changement concernant l'expéditeur et un nœud ayant l'*ID de nœud* approprié.

Sur la voie de gestion unidiffusion, un paquet TNR est envoyé par l'entité feuille (LE) qui change de parent au moyen d'un changement d'arborescence ou d'une participation à une arborescence. Il est également utilisé par un nœud pour se séparer de son enfant après en avoir détecté la défaillance. Il est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille;
- port source: numéro de port local de l'entité feuille;
- adresse IP de destination: adresse IP du propriétaire local LO;
- port de destination: numéro de port de groupe.

Le paquet TNR contient l'en-tête de base de 16 octets et l'élément Informations de changement d'arborescence de 8 octets, comme indiqué dans la Figure 34.

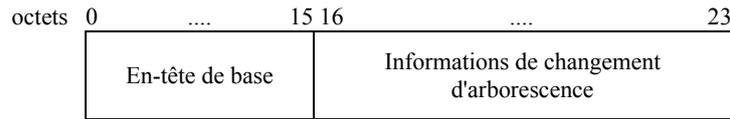


Figure 34 – Paquet TNR

L'en-tête de base du paquet TNR doit être codé comme suit:

*Élément suivant:* '1001' (élément Informations de changement d'arborescence).

*CT:* '11'.

*Type de paquet:* '0010 0001' (TNR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local de l'entité feuille (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro de séquence de paquet.

*Longueur de capacité utile:* '8'.

*F:* F est mis à '0' s'il notifie un changement de nœud parent, ou mis à '1' s'il notifie le détachement du nœud enfant.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Informations de changement d'arborescence de 8 octets doit être codé comme suit:

*Élément suivant:* '0000'.

*ID de nœud:* ID du nouveau nœud parent lorsque F est '0' ou ID du nœud enfant qui est détaché lorsque F est '1'.

### 8.3.28 Confirmation de notification de changement d'arborescence (TNC, *tree change notification confirm*)

Le paquet TNC est envoyé en réponse à un paquet TNR. Le propriétaire local (LO) envoie un paquet TNC à un nœud qui a envoyé le paquet TNR aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire local;
- port source: numéro de port de groupe;
- adresse IP de destination: adresse IP de l'entité feuille;
- port de destination: port local de l'entité feuille.

Le paquet TNC contient uniquement l'en-tête de base de 16 octets. L'en-tête de base du paquet TNC doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0010 0010' (TNC).

*Total de contrôle:* à calculer.

*Port source:* numéro de port de groupe (ou ID de connexion).

*Port de destination:* numéro de port local de l'entité feuille (ou ID de connexion).

*PSN:* valeur reprise du champ PSN du paquet de demande correspondant.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

### 8.3.29 Demande de changement d'arborescence de gestion (CCR, *control tree change request*)

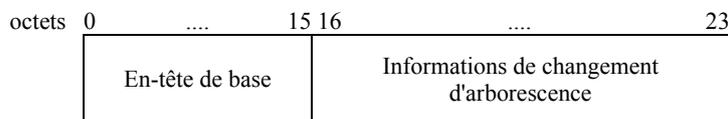
Le paquet CCR est envoyé par le propriétaire local (LO) pour changer l'arborescence de gestion de l'entité feuille (LE). Sur la voie de gestion unidiffusion, un propriétaire local envoie un paquet CCR à une entité feuille qui devrait changer ses informations d'arborescence de gestion. Il est envoyé aux adresses source et de destination suivantes:

- adresse IP source: adresse IP du propriétaire local;

## ISO/CEI 14476-5:2008 (F)

- port source: numéro de port local du propriétaire local;
- adresse IP de destination: adresse IP du nœud cible (entité feuille);
- port de destination: numéro de port de groupe.

Le paquet CCR contient l'en-tête de base de 16 octets et l'élément Informations de changement d'arborescence de 8 octets, comme indiqué dans la Figure 35.



**Figure 35 – Paquet CCR**

L'en-tête de base du paquet CCR doit être codé comme suit:

*Élément suivant:* '1001' (élément Informations de changement d'arborescence).

*CT:* '11'.

*Type de paquet:* '0010 1000' (CCR).

*Total de contrôle:* à calculer.

*Port source:* numéro de port local du propriétaire local (ou ID de connexion).

*Port de destination:* numéro de port de groupe (ou ID de connexion).

*PSN:* numéro de séquence de paquet.

*Longueur de capacité utile:* '8'.

*ID de jeton:* ID de jeton de l'utilisateur expéditeur SU correspondant.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

L'élément Informations de changement d'arborescence de 8 octets doit être codé comme suit:

*Élément suivant:* '0000'.

*ID de nœud:* représente le nouveau nœud parent dans l'arborescence de gestion pour l'utilisateur expéditeur SU défini par l'*ID de jeton* de l'en-tête de base.

### 8.3.30 Confirmation de changement d'arborescence de gestion (CCC, *control tree change confirm*)

Le paquet CCC est envoyé en réponse à un paquet CCR. Une entité feuille (LE) envoie un paquet CCC au propriétaire local (LO) qui a envoyé le paquet CCR aux adresses source et de destination suivantes:

- adresse IP source: adresse IP de l'entité feuille;
- port source: numéro de port local de l'entité feuille;
- adresse IP de destination: adresse IP du propriétaire local;
- port de destination: numéro de port de groupe.

Le paquet CCC contient uniquement l'en-tête de base de 16 octets, qui doit être codé comme suit:

*Élément suivant:* '0000'.

*CT:* '11'.

*Type de paquet:* '0010 1001' (CCC).

*Total de contrôle:* à calculer.

*Port source:* numéro de port de groupe (ou ID de connexion).

*Port de destination:* numéro de port local du propriétaire local (ou ID de connexion).

*PSN:* valeur reprise du champ PSN du paquet de demande correspondant.

*ID de jeton:* ID de jeton de l'utilisateur expéditeur SU correspondant.

Tous les champs autres que ceux qui sont définis plus haut seront mis à '0' et ne seront pas pris en compte par le destinataire.

## 9 Procédures

Le présent paragraphe décrit les procédures protocolaires ECTP-5. Avant la création d'une connexion multidiffusée n-plex, les informations d'adresse ci-après devraient être annoncées aux participants potentiels, à savoir les utilisateurs du service TS.

- a) adresse IP multidiffusion de groupe;
- b) numéro de port de groupe;
- c) adresse IP du propriétaire de la connexion TC;
- d) adresse IP d'un propriétaire local correspondant (destinée à l'entité feuille uniquement).

Ces informations peuvent être annoncées aux participants potentiels par un mécanisme de signalisation hors bande tel qu'une annonce sur le Web. En conséquence, l'utilisateur du service TS potentiel devrait être en mesure de rattacher l'adresse IP et le port de groupe afin de recevoir le paquet CR du propriétaire de la connexion TC. Un utilisateur du service TS potentiel qui est un participant tardif devrait également envoyer un paquet JR au propriétaire de la connexion TC.

### 9.1 Gestion de connexion

#### 9.1.1 Création de connexion

Une connexion multidiffusée n-plex débutera lorsque le propriétaire de la connexion TC (TCN) est activé en vue de la gestion des informations et des jetons au cours de la session.

Si le propriétaire de la connexion TC a connaissance d'une liste de participants avant le début de la session, il lance une phase de création de connexion en envoyant un paquet CR au groupe à l'adresse IP multidiffusion et au port de groupe.

Les opérations générales de création de connexion sont indiquées dans la Figure 36:

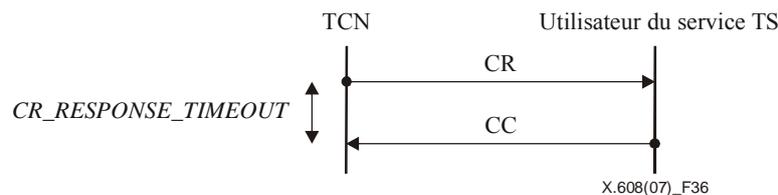


Figure 36 – Procédure de création de connexion

Le paquet CR contient des informations générales sur l'élément Connexion, par exemple l'option de configuration d'arborescence (TCO) et la taille maximale de segment (MSS).

Si tous les paquets CC n'arrivent pas dans le délai d'attente des réponses au paquet CR (*CR\_RESPONSE\_TIMEOUT*), le propriétaire de la connexion TC envoie à nouveau un paquet CR. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet CR (*CR\_MAX\_RETRY*). Si le propriétaire de la connexion TC n'a pas reçu de paquet CC de tous les utilisateurs du service TS figurant dans la liste de participants, il abandonne la procédure de création de connexion et met fin à la connexion multidiffusée n-plex en envoyant le paquet CT (fin de connexion) à un groupe. S'il n'y a aucun participant prédéterminé avant le début de la session, le propriétaire de la connexion TC lance la phase de transmission de données sans opérations de création de connexion.

#### 9.1.2 Participation tardive

Certains des participants potentiels peuvent se joindre à la connexion multidiffusée n-plex en tant que participants tardifs. Les opérations générales concernant un participant tardif sont indiquées dans la Figure 37.

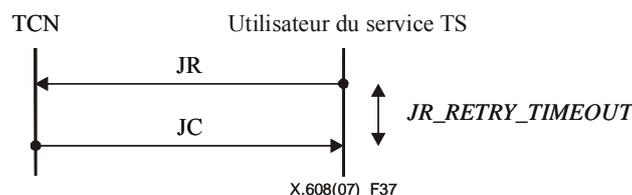


Figure 37 – Procédure de participation tardive

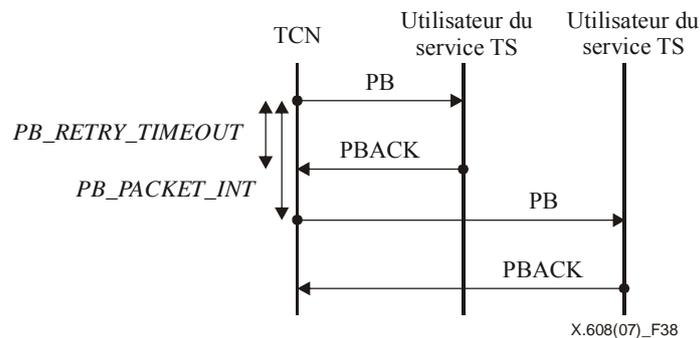
L'utilisateur du service TS qui est un participant tardif envoie un paquet JR au propriétaire de la connexion. En réponse au paquet JR, le propriétaire de la connexion TC envoie un paquet JC à l'utilisateur du service TS. Le paquet TJ devrait indiquer si la demande est acceptée ou non au moyen de l'indicateur *F* de l'en-tête de base.

Si le paquet JC n'arrive pas dans le délai de réessai pour le paquet JR (*JR\_RETRY\_TIMEOUT*), le participant tardif envoie à nouveau le paquet JR. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet JR (*JR\_MAX\_RETRY*). Si le nœud n'a reçu aucun paquet JC, il abandonne la procédure de participation tardive et met fin à la connexion multidiffusée n-plex.

**9.1.3 Mise à jour de la connexion**

Une connexion multidiffusée n-plex est mise à jour au moyen des paquets PB et PBACK. Au cours de la session, le propriétaire de la connexion TC envoie périodiquement, à chaque intervalle d'envoi du paquet PB (*PB\_PACKET\_INT*), des paquets PB à un des utilisateurs du service TS. L'utilisateur du service TS correspondant devrait répondre par le paquet PBACK. La méthode de sélection de l'utilisateur du service TS pour l'envoi des paquets PB devrait être conçue de façon à couvrir chaque utilisateur du service TS pendant la session, par exemple par la méthode de l'ordonnancement cyclique (*round robin*).

La Figure 38 montre l'opération de gestion des membres au moyen des paquets PB et PBACK.

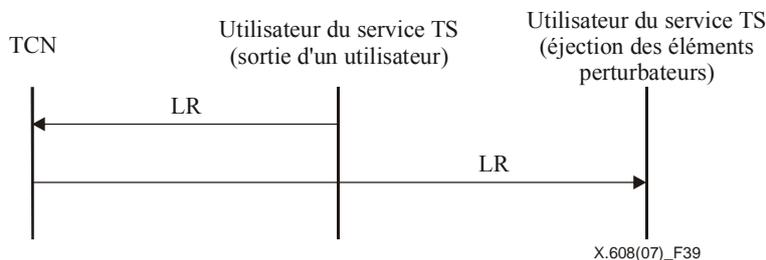


**Figure 38 – Mise à jour de la connexion au moyen des paquets PB et PBACK**

Si le paquet PBACK n'arrive pas dans le délai de réessai pour le paquet PB *PB\_RETRY\_TIMEOUT*, le propriétaire de la connexion TC envoie à nouveau le paquet PB. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet PB *PB\_MAX\_RETRY*. Si le propriétaire de la connexion TC n'a reçu aucun paquet PBACK, il force l'utilisateur du service TS à quitter la connexion en envoyant un message LR (éjection des éléments perturbateurs).

**9.1.4 Sortie d'un utilisateur**

La Figure 39 montre les opérations relatives à une sortie activée par l'utilisateur et à l'éjection des éléments perturbateurs.



**Figure 39 – Procédures de sortie d'un utilisateur et d'éjection des éléments perturbateurs**

Dans le cas de la sortie d'un utilisateur, l'utilisateur du service TS enverra un message LR au propriétaire de la connexion TC (TCN). Dans le cas de l'éjection des éléments perturbateurs, le propriétaire de la connexion TC demandera à l'utilisateur du service TS concerné de quitter la connexion. Dans les deux cas, le message LR n'exige pas de message de confirmation correspondant.

L'éjection des éléments perturbateurs est appliquée à l'utilisateur du service TS qui n'a pas donné de réponse dans un délai donné dans les opérations PB et PBACK destinées à la mise à jour de la connexion.

### 9.1.5 Fin de la connexion

Dans le protocole ECTP-5, le propriétaire de la connexion TC peut aussi mettre fin à la connexion lorsqu'il en a ainsi décidé. Il le fait en envoyant un message CT au groupe. La Figure 40 indique les opérations de fin de connexion. Les paquets CT n'exigent aucun message de confirmation.

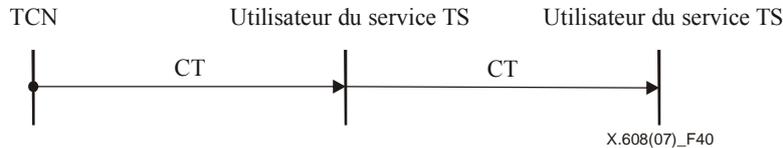


Figure 40 – Procédure de fin de connexion

## 9.2 Gestion d'arborescence logique

### 9.2.1 Participation à une arborescence intra-groupe

Chaque entité feuille (LE) devrait participer à une arborescence intra-groupe pour la protection contre les erreurs après une participation par création de connexion initiale ou par participation tardive.

Une entité feuille lance une procédure de participation à une arborescence intra-groupe en envoyant un paquet TJ au propriétaire local (LO) correspondant, qui répond par un paquet TC. Celui-ci devrait indiquer si la demande de participation à une arborescence est acceptée ou non au moyen de l'indicateur F de l'en-tête de base. Les opérations générales concernant la participation à une arborescence intra-groupe sont indiquées dans la Figure 41:

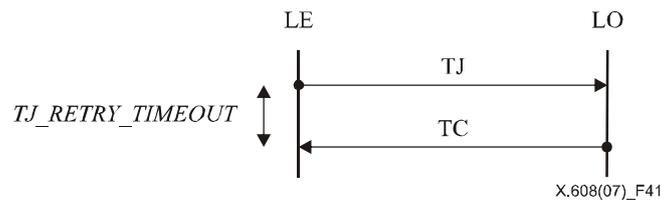


Figure 41 – Procédure de participation à une arborescence intra-groupe

Si aucun paquet TC n'est reçu du propriétaire local en réponse à un paquet TJ dans le délai de réessai pour le paquet TJ *TJ\_RETRY\_TIMEOUT*, le nœud envoie à nouveau un paquet TJ. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TJ *TJ\_MAX\_RETRY*. Si le nœud ne reçoit pas de paquet TC, il abandonne la procédure de participation à une arborescence et renvoie à l'application une erreur avec les informations d'état.

Cette procédure peut aussi être utilisée entre des entités feuilles au changement d'arborescence par adaptation d'arborescence logique ainsi qu'il est décrit au § 9.2.4.

### 9.2.2 Participation à une arborescence inter-groupes

À réception de paquets TSR, si un propriétaire local détecte un nouveau propriétaire local qui a un ou plusieurs utilisateurs expéditeurs SU, il devrait se joindre à une arborescence inter-groupes dont la racine est le nouveau propriétaire local (LO).

Pour être un utilisateur expéditeur SU, une entité feuille (LE) devrait notifier au propriétaire de la connexion TC son propriétaire local pendant la procédure d'obtention ou de cession de jeton. Le propriétaire de la connexion TC met à jour les propriétaires locaux et leur fournit les informations en question en envoyant périodiquement des paquets TSR. Le paquet TSR contient l'élément Informations sur le propriétaire local qui comprend une liste d'ID de jetons de chaque groupe local.

Au moyen de ces informations, un propriétaire local se joint à chaque arborescence inter-groupes dont la racine est le propriétaire local en envoyant un paquet TJ à chaque propriétaire local qui a un ou plusieurs utilisateurs SU dans son groupe local.

Chaque propriétaire local qui est la racine de l'arborescence inter-groupes répond ensuite par un paquet TC. Celui-ci devrait indiquer si la demande de participation à une arborescence est acceptée ou non au moyen de l'indicateur F de l'en-tête de base. Les opérations générales concernant la participation à une arborescence inter-groupes sont indiquées dans la Figure 42.

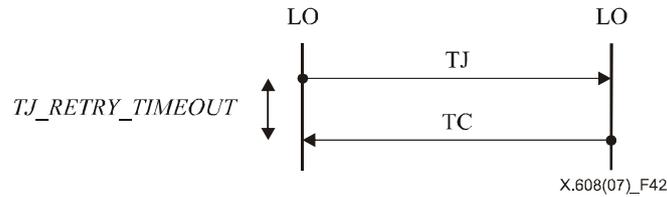


Figure 42 – Procédure de participation à une arborescence inter-groupes

Si le paquet TC n'arrive pas dans le délai de réessai pour le paquet TJ *TJ\_RETRY\_TIMEOUT*, l'expéditeur du paquet TJ peut essayer de l'envoyer à nouveau. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet *TJ\_MAX\_RETRY*. S'il n'a pas reçu de paquet TC, le nœud abandonne la procédure de participation à une arborescence et met fin à la connexion multidiffusée n-plex.

### 9.2.3 Sortie d'arborescence logique

Avant de quitter une session (sortie d'un utilisateur) ou de changer de nœud parent au moyen de la procédure d'adaptation d'arborescence logique, une entité feuille (LE) dépourvue de nœuds enfants devrait quitter l'arborescence logique (arborescence intra-groupe) en envoyant un paquet TLR à son nœud parent. Ce dernier supprime alors l'entité feuille de la liste de ses enfants et répond par un paquet TLC.

Les opérations générales concernant la sortie de l'arborescence intra-groupe sont indiquées dans la Figure 43:

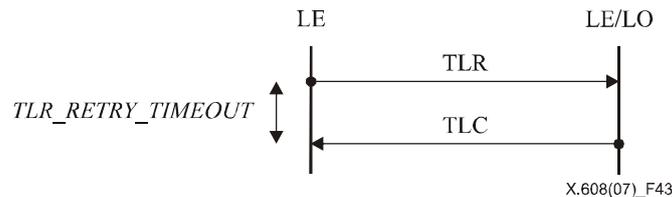
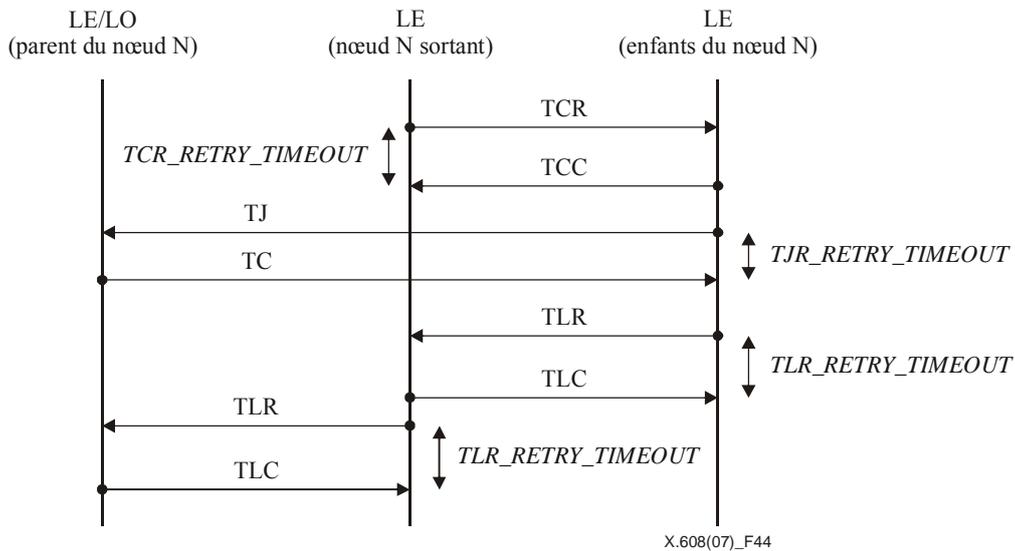


Figure 43 – Procédure de sortie d'une arborescence intra-groupe

Lorsqu'une entité feuille pourvue d'un ou de plusieurs enfants quitte une session, elle devrait rattacher ses enfants à son parent en envoyant un message TCR à ses enfants avant sa sortie. Un nœud enfant qui reçoit le message TCR répond par un message TCC. En outre, les nœuds enfants envoient des messages TJ à leur nouveau parent pour se joindre en tant qu'enfants. Après s'être joints au nouveau parent, les nœuds enfants envoient des messages TLR à leur parent précédent. Ensuite, l'entité feuille sortante répond à chaque message TLR par des messages TLC. Après avoir répondu aux messages TLR émanant de tous ses nœuds enfants, l'entité feuille sortante peut quitter l'arborescence en envoyant un message TLR à son parent. Les opérations générales concernant la sortie d'une arborescence intra-groupe sont indiquées dans la Figure 44:

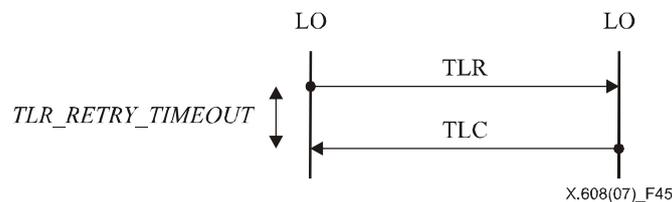


X.608(07)\_F44

Figure 44 – Procédure de sortie d'une arborescence intra-groupe pour un nœud N ayant des enfants

Un propriétaire local devrait quitter toutes les arborescences inter-groupes s'il n'a pas d'enfant dans son groupe local et s'il n'a pas d'application pour utiliser les données de la session. Si un propriétaire local n'a pas d'utilisateur expéditeur SU, tous les autres propriétaires locaux devraient se détacher de l'arborescence inter-groupes dont la racine est le propriétaire local.

Un propriétaire local peut quitter une arborescence inter-groupes en envoyant un paquet TLR à son propriétaire local parent. Le propriétaire local parent qui reçoit le paquet TLR supprime ensuite le propriétaire local de la liste de ses enfants et répond au propriétaire local par un paquet TLC. Les opérations générales concernant la sortie d'une arborescence inter-groupes sont indiquées dans la Figure 45:



X.608(07)\_F45

Figure 45 – Procédure de sortie d'une arborescence inter-groupes

Si le message TLC de réponse n'est pas arrivé dans le délai de réessai pour le paquet TLR *TLR\_RETRY\_TIMEOUT*, l'entité feuille ou le propriétaire local peut envoyer à nouveau le message TLR à l'entité feuille ou au propriétaire local parent. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TLR *TLR\_MAX\_RETRY*. Si le nœud ne reçoit pas de paquet TLC, il abandonne la procédure de sortie d'arborescence et se détache de son parent.

### 9.2.4 Adaptation d'arborescence logique

Avec l'option TCO '10', les arborescences intra-groupe peuvent devenir des arborescences à plusieurs niveaux proches des arborescences de routage multidiffusion sous-jacentes par comparaison des configurations de pertes des nœuds parents et enfants de l'arborescence. Un nœud peut déterminer la relation qui l'unit à ses enfants en comparant leurs suites binaires d'erreurs.

Pour décrire le mécanisme d'adaptation d'arborescence logique, on définit comme suit trois opérateurs relationnels entre deux nœuds d'une arborescence intra-groupe.

$$B(N) = B(M)$$

Le nœud N et le nœud M ont la relation  $B(N) = B(M)$  (c'est-à-dire que le nœud N est potentiellement égal au nœud M) si et seulement si  $B_k(N) = B_k(M)$  pour tous les  $k = 1, 2, \dots, n$ . Ici,  $B_k(N)$  désigne le k-ième bit (à partir de la gauche) de la suite binaire du nœud N et  $B(N)$  est une chaîne de bits  $B_1(N), B_2(N), \dots, B_n(N)$ , la longueur de suite binaire étant supposée être n.

$B(N) \supset B(M)$

Le nœud  $N$  et le nœud  $M$  ont la relation  $B(N) \supset B(M)$  (c'est-à-dire que le nœud  $N$  est un parent potentiel du nœud  $M$ ) si et seulement si  $B_k(N) \geq B_k(M)$  pour tous les  $k = 1, 2, \dots, n$ , mais sans que  $B(N) = B(M)$ .

$B(N) \subset B(M)$

De même,  $B(N) \subset B(M)$  (c'est-à-dire que le nœud  $N$  est un enfant potentiel du nœud  $M$ ) si et seulement si  $B_k(N) \leq B_k(M)$  pour tous les  $k = 1, 2, \dots, n$ , mais sans que  $B(N) = B(M)$ .

Dans la Figure 46,  $ERROR\_BITMAP(N)$  représente un message contenant la suite binaire d'erreurs du nœud  $N$ . Le message  $TDR(N)$  est créé par un nœud qui ne peut pas déterminer la position du nœud  $N$ . Ce message contient  $ERROR\_BITMAP(N)$  en vue d'autres délégations.  $TCR(N)$  signifie qu'un nœud recevant le message devrait se joindre au nœud  $N$  en tant qu'enfant.  $Parent(N)$  et  $enfant(N)$  représentent le nœud parent et le nœud enfant du nœud  $N$ , respectivement. Le pseudo-code ci-après décrit l'algorithme d'adaptation d'arborescence logique.

```
// node D receives a message m;
case (m is ERROR BITMAP(N))
    if (B(D) ⊂ B(N))
        send a TDR(N) message to parent(D);
    else
        if (∃ a child C such that B(N) ⊃ B(C))
            send a TDR(C) message to N;
        else if (∃ a child C such that B(C) ⊃ B(N))
            send a TDR(N) message to C;
        else
            do nothing;
case (m is TDR(N))
    if (B(D) ⊂ B(N))
        send a TDR(N) message to parent(D)
    else if (B(D) = B(N))
        send a TCR(D) message to N;
    else
        if (∃ a child C such that B(N) ⊃ B(C))
            send a TDR(C) message to N;
            send a TCR(D) message to N;
        else if (∃ a child C such that B(C) ⊃ B(N))
            send a TDR(N) message to C;
        else
            send a TCR(D) message to N;
```

**Figure 46 – Pseudo-code concernant l'adaptation d'arborescence logique**

Le message  $TDR(N)$  est transmis au parent potentiel du nœud  $N$ . Le même processus se répète en vers le haut et vers le bas de l'arborescence intra-groupe jusqu'à ce que le nœud  $N$  trouve la position appropriée. La Figure 47 montre une procédure permettant de déléguer un nœud au parent potentiel.

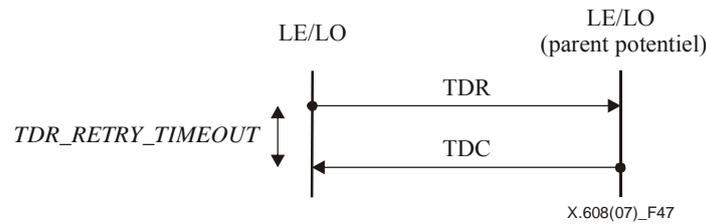


Figure 47 – Procédure de délégation d'arborescence

Si le paquet TDC n'arrive pas dans le délai de réessai pour le paquet TDR  $TDR\_RETRY\_TIMEOUT$ , l'expéditeur du paquet TDR peut essayer d'envoyer à nouveau le paquet TDR. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TDR  $TDR\_MAX\_RETRY$ . Si le nœud ne reçoit pas de paquet TDC, il abandonne la procédure de délégation d'arborescence.

Si un nœud peut déterminer une position appropriée pour le nœud  $N$ , il envoie un message TCR au nœud  $N$ . A réception du message TCR, le nœud  $N$  répond par un message TCC pour confirmer la réception du message TCR émanant de son nouveau parent. Le nœud  $N$  se joint au nouveau parent en envoyant un message TJ. Ensuite, le nœud  $N$  quitte son ancien parent. En quittant son ancien parent après s'être joint à un nouveau parent, le nœud  $N$  peut continuellement récupérer les paquets perdus provenant de l'ancien parent au milieu du processus de délégation et de changement d'arborescence.

Si le paquet TCC n'arrive pas dans le délai de réessai pour le paquet TCR  $TCR\_RETRY\_TIMEOUT$ , l'expéditeur du paquet TCR peut essayer de l'envoyer à nouveau. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TCR  $TCR\_MAX\_RETRY$ . S'il ne reçoit pas de paquet TCC, le nœud abandonne la procédure de changement d'arborescence.

Les opérations générales concernant la procédure de changement d'arborescence sont indiquées dans la Figure 48:

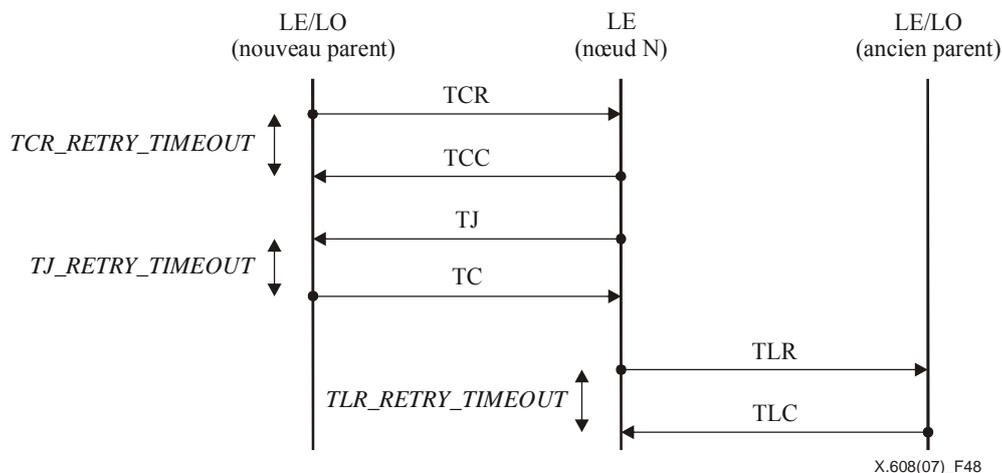
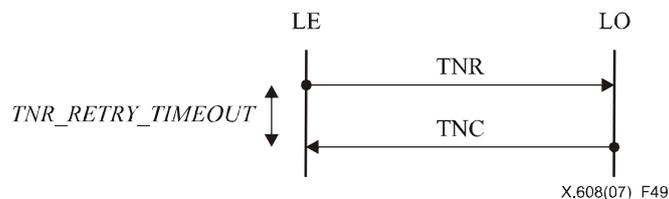


Figure 48 – Procédure de changement d'arborescence

### 9.2.5 Notification de changement d'arborescence logique

Le propriétaire local (LO) devrait gérer son arborescence intra-groupe afin de créer des arborescences de gestion pour les utilisateurs SU. Ainsi, il devrait être informé du changement d'arborescence intra-groupe causé par la procédure de participation à l'arborescence logique initiale, par le mécanisme d'adaptation d'arborescence logique ou par une adaptation à la défaillance du nœud.

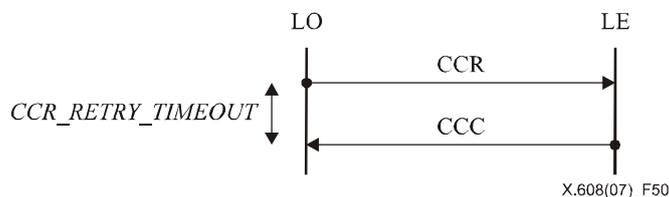
Le message TNR est envoyé au propriétaire local par les entités feuilles à réception du message TJC après détection d'une défaillance de son enfant. Les opérations générales concernant la procédure de changement d'arborescence sont indiquées dans la Figure 49:



**Figure 49 – Procédure de notification de changement d'arborescence**

Si le paquet TNC n'arrive pas dans le délai de réessai pour le paquet TNR  $TNR\_RETRY\_TIMEOUT$ , l'expéditeur du paquet TNR peut essayer d'envoyer le paquet TNR à nouveau. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TNR  $TNR\_MAX\_RETRY$ . Si le nœud ne reçoit pas de paquet TNC, il abandonne la procédure de notification de changement d'arborescence, renvoie à l'application une erreur avec les informations d'état et met fin à la connexion multidiffusée n-plex.

A réception du message TNR par le propriétaire local, celui-ci devrait mettre à jour ses informations d'arborescence logique. Avec la conclusion de l'arborescence logique, il examine les arborescences de gestion pour notifier à chaque entité feuille qu'elle devrait changer la relation parent-enfant dans l'arborescence de gestion au moyen du message CCR.



**Figure 50 – Procédure de changement d'arborescence de gestion**

Si le paquet CCR n'arrive pas dans le délai de réessai pour le paquet CCR  $CCR\_RETRY\_TIMEOUT$ , le propriétaire local du paquet CCR peut essayer d'envoyer le paquet CCR à nouveau. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet CCR  $CCR\_MAX\_RETRY$ . Si le propriétaire local n'a pas reçu de paquet CCC, il abandonne la procédure de changement d'arborescence de gestion et envoie un paquet LR au destinataire correspondant.

### 9.2.6 Mise à jour d'arborescence logique

Pour mettre à jour une arborescence logique, les utilisateurs du service TS se servent des informations obtenues des paquets NACK et ACK. Si un utilisateur du service TS n'a reçu aucun paquet RD après avoir envoyé un nombre de paquets NACK égal au nombre maximal de réessais pour le paquet NACK  $NACK\_MAX\_RETRY$ , il abandonne la procédure de reprise après incident et suppose une défaillance de son parent. Il essaie ensuite de trouver un autre parent approprié en contactant le propriétaire local. Un utilisateur du service TS examine les numéros LSN de ses enfants, et si le numéro LSN d'un nœud enfant a un écart égal à l'écart maximal autorisé pour le numéro LSN  $MAX\_LSN\_LAG$  par rapport à son propre numéro LSN, il suppose une défaillance de l'enfant. Il détache ensuite l'enfant de l'arborescence logique et en informe le propriétaire local.

## 9.3 Transport de données multidiffusées

Sur la voie de transmission des données multidiffusées du protocole ECTP-5, le propriétaire de la connexion TC ou l'utilisateur expéditeur SU peut envoyer des paquets DT multidiffusés au groupe. Lorsque la perte d'un paquet de données est détectée par l'utilisateur du service TS destinataire, la retransmission en vue de la reprise après incident sera effectuée par un nœud parent le long de l'arborescence de gestion.

### 9.3.1 Transmission de données multidiffusées

Le propriétaire de la connexion TC ou l'utilisateur expéditeur SU créera des paquets DT au moyen de la procédure de segmentation. Pour ce faire, l'expéditeur divise un flux de données multidiffusées de l'application en plusieurs paquets DT. Chaque paquet DT a son propre *ID de jeton* et son propre numéro de séquence.

Chaque utilisateur du service TS transmet tous les paquets de données reçus à l'application dans l'ordre appliqué par le propriétaire de la connexion TC. Chaque destinataire rassemble les paquets reçus. Les paquets altérés et perdus sont détectés au moyen d'un total de contrôle et d'un numéro de séquence. Un paquet altéré est aussi considéré comme perdu. Les paquets DT perdus sont récupérés au moyen de la fonction de protection contre les erreurs.

Il convient de noter que les paquets ACK sont créés pour chaque utilisateur expéditeur SU qui est identifié par l'ID de jeton.

Le protocole ECTP-5 utilise la fonction de gestion de flux en se fondant sur une *fenêtre* de taille fixe. La *taille de fenêtre* représente le nombre de paquets de données sans accusé de réception contenus dans la mémoire tampon d'envoi. L'expéditeur peut transmettre au maximum un nombre de paquets de données égal à la *taille de fenêtre*, au débit de transmission de données configuré. Dans le protocole ECTP-5, le débit de transmission des données multidiffusées est régi par les mécanismes de gestion des encombrements fondés sur le débit.

Un nouveau paquet DT est numéroté séquentiellement par l'expéditeur multidiffusion. Le numéro de séquence du paquet DT commence à partir du numéro PSN initial et augmente de '1'. Il permet de détecter les paquets de données perdus par les destinataires. Le numéro PSN initial est créé de façon aléatoire à une valeur différente de '0'. Le numéro de séquence '0' est réservé. Le numéro de séquence de paquet augmente à chaque nouveau paquet DT. Une arithmétique de modulo  $2^{32}$  est utilisée et le numéro de séquence reboucle sur '1' après avoir atteint la valeur  $2^{32} - 1$ .

### 9.3.2 Gestion de la fiabilité pour un transport fiable

- a) Chaque utilisateur du service TS devrait mettre en mémoire tampon le flux de données provenant de chaque utilisateur expéditeur SU pour une reprise après incident des nœuds enfants le long de l'arborescence de gestion.
- b) Lorsqu'un utilisateur du service TS détecte une ou plusieurs pertes de paquets, il demande à son nœud parent la retransmission des paquets perdus le long de l'arborescence de gestion dont la racine est l'utilisateur expéditeur SU correspondant au moyen d'un message de gestion NACK.
- c) Un utilisateur du service TS devrait retransmettre les données en mode unidiffusion RD (données de retransmission) lorsqu'il a reçu un message NACK d'un nœud enfant le long de l'arborescence de gestion.
- d) Un utilisateur du service TS devrait accuser réception des paquets DT reçus en envoyant un message ACK au nœud parent le long de l'arborescence de gestion.
- e) Les données de la mémoire tampon gérée par l'utilisateur du service TS peuvent être émises lorsque tous les nœuds enfants le long de l'arborescence de gestion en ont accusé réception au moyen d'un message ACK.
- f) Tous les utilisateurs SU devraient mettre en mémoire tampon un ensemble de données de son flux de données même si les nœuds enfants en ont accusé réception. Cela permet une reprise après incident pour un utilisateur du service TS qui n'a pas remédié aux pertes de paquets émanant de son nœud parent le long de l'arborescence de gestion. L'ensemble de données du flux de données est déterminé par la sémantique définie par l'application dans la couche supérieure.

#### 9.3.2.1 Détection des erreurs

Le champ Total de contrôle de l'en-tête de base permet de détecter une altération des paquets et le champ *PSN* permet de détecter une perte de paquets. A réception d'un paquet, chaque destinataire examine le total de contrôle. Si ce champ n'est pas valide, le paquet est réputé altéré et doit être ignoré. Une altération est traitée comme une perte. La perte peut être détectée comme étant un écart entre deux numéros de séquence consécutifs concernant les paquets DT. Les paquets NACK sont utilisés pour les demandes de retransmission lors de la détection des pertes.

#### 9.3.2.2 Reprise après incident en mode unidiffusion au moyen des paquets NACK et RD

La Figure 51 montre les opérations de reprise après incident sur la voie de gestion.

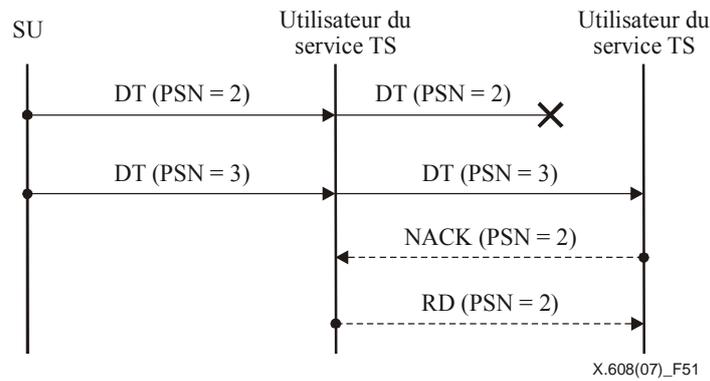


Figure 51 – Procédure de reprise après incident

Si un participant détecte une ou plusieurs pertes de paquets, il transmet immédiatement un paquet NACK au parent le long de l'arborescence de gestion. En réponse au paquet NACK, le parent transmet un ou plusieurs paquets RD à l'enfant en mode unidiffusion.

Si le nœud parent a reçu un paquet NACK demandant une réparation pour un paquet de données déjà diffusé, il répond par un paquet RD pourvu d'un indicateur '1'. L'enfant qui reçoit le paquet RD pourvu d'un indicateur '1' envoie un autre paquet NACK à l'utilisateur expéditeur SU correspondant. Pour cela, l'utilisateur expéditeur SU devrait mettre en mémoire tampon des quantités suffisantes de données pour pouvoir réparer toute perte.

**9.3.2.3 Retransmission de paquets NACK au moyen d'un temporisateur**

Un paquet RD peut ne pas être transmis au demandeur d'une réparation à cause de la perte d'un paquet NACK provenant du demandeur ou d'un paquet RD émanant du parent. Un paquet NACK peut ensuite être retransmis comme le montre la Figure 52:

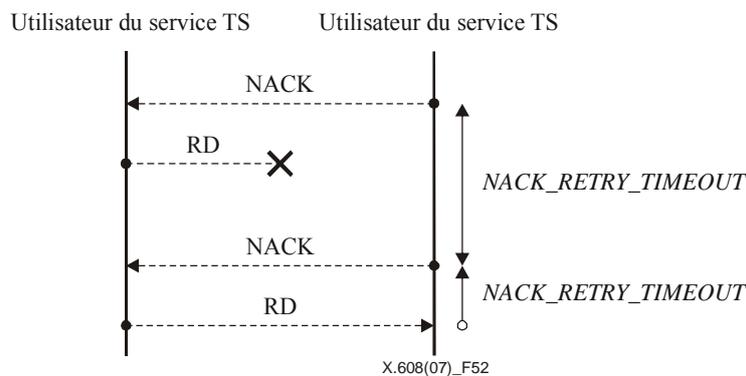


Figure 52 – Procédure de retransmission de paquets NACK

Un paquet NACK peut être retransmis au parent si le demandeur d'une réparation n'a pas reçu le paquet RD correspondant après le délai de réessai pour le paquet NACK *NACK\_RETRY\_TIMEOUT*. Le temporisateur *NACK\_RETRY\_TIMEOUT* est lancé lors de l'envoi de chaque paquet NACK et il est annulé à réception du paquet RD correspondant. Le processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet NACK *NACK\_MAX\_RETRY*. Si le nœud n'a reçu aucun paquet RD, il abandonne la procédure de reprise après incident et suppose une défaillance de son parent. Il essaie ensuite de trouver un autre parent approprié en contactant le propriétaire local (LO).

**9.3.2.4 Création de paquets ACK**

La Figure 53 montre les opérations d'accusé de réception sur la voie de gestion.

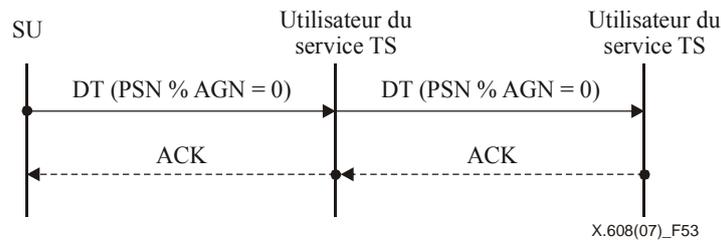


Figure 53 – Procédure de gestion des paquets ACK

Chaque enfant crée un paquet ACK en fonction de la fréquence de création de paquets *ACK\_GENERATION\_NUM* (*AGN*). Chaque fois que le nombre de paquets est égal à *AGN*, il examine l'état de transmission des paquets entre le paquet dont il a été fait précédemment accusé réception et le dernier paquet reçu. Si tous les paquets de la série ont bien été reçus, il envoie un paquet ACK à son parent.

Chaque enfant envoie un paquet ACK à son parent, si le numéro *PSN* d'un paquet DT modulo *AGN* est égal à zéro, c'est-à-dire si:

$$PSN \% AGN = 0$$

Supposons que *AGN* = 8; l'enfant crée un paquet ACK pour les paquets DT dont les numéros de séquence sont 8, 16, 24, 32, etc. Cette règle de création de paquets ACK s'applique lorsque les paquets DT ou RD correspondants sont reçus par l'enfant.

#### 9.3.2.5 Accusé de réception de données et regroupement de paquets ACK

Chaque parent utilise des paquets ACK pour rassembler des informations d'état sur la réception de données par les utilisateurs du service TS. Chaque fois qu'un parent reçoit un paquet ACK de l'un de ses enfants, il enregistre et met à jour les informations d'état concernant les paquets effectivement reçus par ses enfants.

Un paquet DT est défini comme "stable" si tous les enfants l'ont reçu. Les paquets DT stables peuvent être émis à partir de la mémoire tampon du parent.

## 9.4 Gestion des jetons

Dans le protocole ECTP-5, un jeton représente le droit d'un utilisateur du service TS de transmettre des données multidiffusées. Chaque utilisateur du service TS désireux de transmettre des données doit obtenir un jeton auprès du propriétaire de la connexion TC (TCN). L'utilisateur du service TS sera un utilisateur expéditeur SU après avoir obtenu un jeton du propriétaire de la connexion TC. Ainsi, ce dernier autorise un utilisateur du service TS à devenir un expéditeur de sorte que les utilisateurs du service TS peuvent effectivement filtrer et rejeter les données multidiffusées envoyées par des utilisateurs non autorisés. Cela étant, il est à noter que l'utilisation de jetons n'assure aucune protection en ce qui concerne la multidiffusion IP.

Un utilisateur expéditeur SU devrait restituer le jeton après la transmission de données.

### 9.4.1 Obtention de jeton

L'utilisateur du service TS peut obtenir un jeton de deux façons: au moyen de la procédure d'obtention de jeton initiée par l'utilisateur du service TS et au moyen de la procédure de cession de jeton initiée par le propriétaire de la connexion TC. Dans la première opération, l'utilisateur du service TS demande d'abord un jeton au propriétaire de la connexion, alors que dans le deuxième cas, le propriétaire de la connexion TC donne d'abord un jeton à un utilisateur potentiel du service TS.

La Figure 54 montre les opérations concernant l'obtention et la cession de jeton.

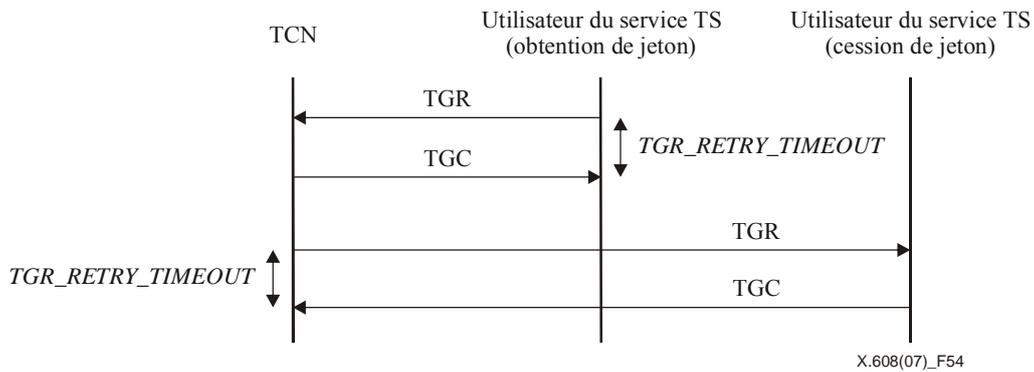


Figure 54 – Procédures d'obtention et de cession de jeton

Pour obtenir un jeton dans l'opération d'obtention de jeton, un utilisateur du service TS envoie un message TGR au propriétaire de la connexion TC et attend ensuite le message TGC correspondant. En réponse au paquet TGR, le propriétaire de la connexion TC devrait envoyer un message TGC à l'utilisateur du service TS. Ce message devrait indiquer si la demande est acceptée ou non au moyen de l'indicateur *F* de l'en-tête de base. En cas d'acceptation, le message contiendra aussi un *ID de jeton* valide dans l'en-tête de base. Si le message TGC de réponse n'est pas arrivé dans le délai de réessai pour le paquet TGR *TGR\_RETRY\_TIMEOUT*, l'utilisateur du service TS peut à nouveau envoyer le message TGR au propriétaire de la connexion. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TGR *TGR\_MAX\_RETRY*. Si le nœud ne reçoit pas de paquet TGC, il abandonne la procédure d'obtention de jeton et renvoie à l'application une erreur avec les informations d'état.

Dans l'opération de cession de jeton, le propriétaire de la connexion TC enverra un message TGR à un utilisateur du service TS. Le message TGR devrait contenir l'*ID de jeton* dans l'en-tête de base.

L'utilisateur du service TS (c'est-à-dire l'utilisateur expéditeur SU) devrait répondre par le message TGC qui met l'indicateur *F* à '1' (acceptation). Si le message TGC de réponse n'est pas arrivé en provenance de l'utilisateur expéditeur SU dans le délai de réessai pour le paquet TGR *TGR\_RETRY\_TIMEOUT*, le propriétaire de la connexion TC peut à nouveau envoyer le message TGR à l'utilisateur expéditeur SU. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TGR *TGR\_MAX\_RETRY*. Si le propriétaire de la connexion TC ne reçoit pas de paquet TGC, il abandonne la procédure de cession de jeton et exclut l'utilisateur du service TS de la liste des utilisateurs SU valides pour que les paquets TSR suivants n'incluent pas cet utilisateur.

#### 9.4.2 Restitution de jeton

A la fin de la transmission de données, l'utilisateur expéditeur SU peut restituer le jeton au propriétaire de la connexion TC. L'utilisateur expéditeur SU peut restituer son jeton au propriétaire de la connexion TC de deux façons: au moyen de la procédure de restitution de jeton initiée par l'utilisateur du service TS et au moyen de la procédure de retrait de jeton initiée par le propriétaire de la connexion TC. Dans la première opération, l'utilisateur du service TS envoie le paquet TRR au propriétaire de la connexion, alors que dans le deuxième cas, le propriétaire de la connexion TC enverra d'abord le message TRR à un utilisateur du service TS.

La Figure 55 montre les opérations concernant la restitution et le retrait de jeton.

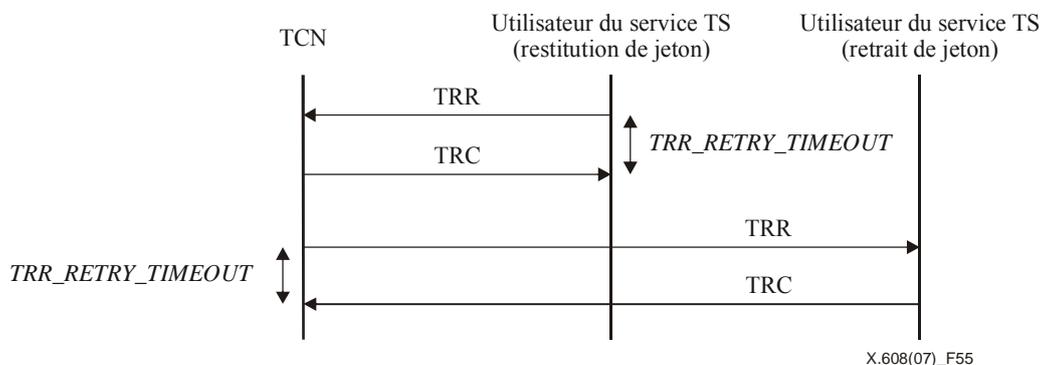


Figure 55 – Procédures de restitution et de retrait de jeton

Dans le cas de la restitution de jeton, l'utilisateur expéditeur SU envoie un message TRR au propriétaire de la connexion TC. Ce dernier répond ensuite par le message TRC. Par contre dans le cas du retrait de jeton, le propriétaire de la connexion TC peut forcer l'utilisateur expéditeur SU concerné à restituer le jeton en envoyant un message TRR. Si le message TRC de réponse n'est pas arrivé dans le délai de réessai pour le paquet TRR *TRR\_RETRY\_TIMEOUT*, le message TRR peut à nouveau être envoyé. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TRR *TRR\_MAX\_RETRY*.

Dans le cas de la restitution de jeton, si l'utilisateur du service TS ne reçoit pas de paquet TRC, il abandonne la procédure. Dans le cas du retrait de jeton, si le propriétaire de la connexion TC ne reçoit pas de paquet TRC, il abandonne la procédure, supprime le jeton de la liste des jetons valides et envoie rapidement le message TSR mis à jour pour aviser les participants.

### 9.4.3 Rapport sur l'état des jetons

Le propriétaire de la connexion TC indique l'état des ID de jetons valides dans la connexion en envoyant un paquet TSR. Ce paquet est transmis par le propriétaire de la connexion TC lorsqu'un nouvel utilisateur expéditeur SU arrive, lorsqu'un utilisateur expéditeur SU existant arrête la transmission de données, ou périodiquement à un intervalle égal à l'intervalle d'envoi du paquet TSR *TSR\_PACKET\_INT* pour la mise à jour des jetons.

La Figure 56 montre les opérations concernant le rapport sur l'état des jetons.

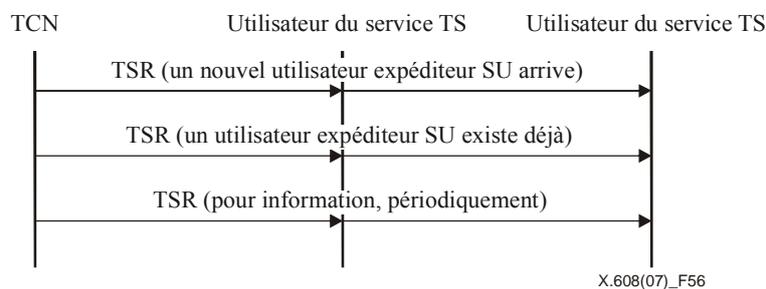


Figure 56 – Procédure concernant le rapport sur l'état des jetons

Si un utilisateur du service TS reçoit un paquet DT d'un utilisateur expéditeur SU qui ne figure pas dans liste des jetons, il demande au propriétaire de la connexion TC de mettre à jour cette liste en envoyant un message TSRR (demande TSR) au propriétaire de la connexion. A réception du message TSRR, le propriétaire de la connexion TC répond par un message TSR comprenant la liste de jetons mise à jour. Si un paquet TSR n'arrive pas dans le délai de réessai pour le paquet TSRR *TSRR\_RETRY\_TIMEOUT*, l'utilisateur TS expéditeur du paquet TSRR peut essayer d'envoyer ce paquet à nouveau. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TSRR *TSRR\_MAX\_RETRY*. Si le nœud n'a pas reçu de paquet TSR, il abandonne la procédure de demande de rapport sur l'état des jetons et ne tient pas compte des paquets DT provenant de l'utilisateur expéditeur SU.

Si un utilisateur du service TS ne reçoit pas le paquet suivant dans le délai d'arrivée du paquet TSR suivant *TSR\_ARRIVAL\_TIMEOUT* après avoir reçu le dernier paquet TSR, il demande au propriétaire de la connexion TC de confirmer que la connexion est valide en lui envoyant un paquet TSRR (demande TSR). A réception du paquet TSRR, le propriétaire de la connexion TC répond par un paquet TSR confirmant la disponibilité de la connexion. Si un paquet TSR n'arrive pas dans le délai de réessai pour le paquet TSRR *TSRR\_RETRY\_TIMEOUT*, l'utilisateur TS expéditeur du paquet TSRR peut essayer d'envoyer ce paquet à nouveau. Ce processus peut se répéter jusqu'à un nombre de fois égal au nombre maximal de réessais pour le paquet TSRR *TSRR\_MAX\_RETRY*. Si le nœud n'a pas reçu de paquet TSR, il abandonne la procédure, renvoie à l'application une erreur avec les informations d'état et met fin à la connexion multidiffusée n-plex.

La Figure 57 montre les opérations concernant la demande de rapport sur l'état des jetons.

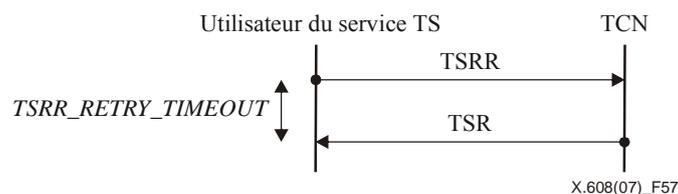


Figure 57 – Procédure de demande de rapport sur l'état des jetons

## 9.5 Mesure du temps RTT

Le temps de transmission aller retour (RTT, *round trip time*) entre un parent et un nœud enfant peut servir à déterminer l'intervalle de réessai pour les paquets de gestion qui sont échangés entre eux. Pour estimer la valeur du temps RTT, l'heure de l'expéditeur est envoyée avec les paquets TJ et NACK. Le nœud recevant le paquet répond ensuite par les paquets TC et RD respectivement qui contiennent la valeur temporelle copiée du paquet reçu.

## 10 Paramètres du système

Le Tableau 4 indique les paramètres du système ECTP-5.

**Tableau 4 – Paramètres du système ECTP-5**

Nom	Description
<i>ACK_GENERATION_NUM</i>	Ecart minimal entre les numéros PSN pour l'envoi du paquet ACK suivant
<i>CCR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet CCR
<i>CCR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet CCR
<i>CR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet CR
<i>CR_RESPONSE_TIMEOUT</i>	Temps d'attente des réponses au paquet CR provenant des utilisateurs du service TS
<i>JR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet JR
<i>JR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet JR
<i>MAX_LSN_LAG</i>	Ecart maximal autorisé pour le numéro LSN du nœud enfant
<i>MAX_SEGMENT_SIZE</i>	Taille maximale du segment de données utilisateur
<i>NACK_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet NACK
<i>NACK_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet NACK
<i>PB_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet PB
<i>PB_PACKET_INT</i>	Intervalle d'envoi du paquet PB
<i>PB_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet PB
<i>TCR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TCR
<i>TCR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TCR
<i>TD_PACKET_INT</i>	Intervalle d'envoi de paquets de données de test
<i>TD_PACKET_NUM</i>	Nombre de paquets de données de test pour un processus d'adaptation d'arborescence logique
<i>TD_PACKET_SIZE</i>	Taille de capacité utile du paquet de données de test
<i>TDR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TDR
<i>TDR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TDR
<i>TGR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TGR
<i>TGR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TGR
<i>TJ_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TJ
<i>TJ_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TJ
<i>TLR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TLR
<i>TLR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TLR
<i>TNR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TNR
<i>TNR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TNR
<i>TRR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TRR
<i>TRR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TRR
<i>TSR_ARRIVAL_TIMEOUT</i>	Latence maximale jusqu'à l'arrivée du prochain paquet TSR
<i>TSR_PACKET_INT</i>	Intervalle d'envoi du paquet TSR
<i>TSRR_MAX_RETRY</i>	Nombre maximal de réessais pour le paquet TSRR
<i>TSRR_RETRY_TIMEOUT</i>	Intervalle de réessai pour le paquet TSRR

## Annexe A

### Interfaces de programmation d'application

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

La présente annexe définit les interfaces de programmation d'application (API, *application programming interface*). Les interfaces API décrites dans la présente Recommandation | Norme internationale peuvent être utilisées par les applications qui ont recours aux capacités de transport de la partie 5 du protocole ECTP (Rec. UIT-T X.608 | ISO/CEI 14476-5). Elles sont extraites de l'Annexe B de la Rec. UIT-T X.606.1 | ISO/CEI 14476-2.

#### A.1 Aperçu général

##### A.1.1 Fonctions de l'interface API

Le Tableau A.1 contient un récapitulatif des fonctions API.

**Tableau A.1 – Fonctions de l'interface API**

Nom de la fonction	Description
<code>msocket()</code>	Crée un nouveau connecteur de protocole ECTP-5
<code>mbind()</code>	Associe au connecteur un ensemble d'adresses/de ports locaux et de groupe
<code>mconnect()</code>	Le propriétaire de la connexion TC déclenche la création d'une connexion vers une adresse étrangère spécifiée. L'utilisateur du service TS qui est un participant tardif déclenche un processus de ralliement.
<code>maccept()</code>	Les utilisateurs potentiels du service TS se joignent à la connexion multidiffusée n-plex en acceptant le signal de création d'une connexion envoyé par le propriétaire de la connexion TC
<code>msend()</code>	Envoie les données d'application à un groupe de destinations
<code>mrecv()</code>	Remet les données reçues à l'application. Remet à l'application certains messages d'indication destinés à la commande pendant la phase de transfert de données.
<code>mclose()</code>	Met fin à la connexion et libère le connecteur

##### A.1.2 Utilisation des fonctions API

Pour une séquence type recourant aux fonctions API, prière de se reporter au § B.1.2 de la Rec. UIT-T X.606.1 | ISO/CEI 14476-2.

#### A.2 Fonctions API du protocole ECTP-5

##### A.2.1 `msocket()`

Pour utiliser le protocole défini dans la présente Recommandation | Norme internationale, une application DOIT tout d'abord invoquer la fonction `msocket`, qui spécifie le type de protocole de communication souhaité tel que le protocole ECTP utilisant le système IPv4 ou le protocole ECTP utilisant le système IPv6.

```
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 A.2;
- *type*: spécifie le type de connecteur et l'une des constantes représentées dans le Tableau A.3;
- *protocol*: est mis à 0.

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

Family	Description
AF_INET	Protocoles IPv4
AF_INET6	Protocoles IPv6

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

Type	Description
SOCK_ECTP5	Connecteur pour le protocole ECTP, partie 5

La fonction `msocket` renvoie un descripteur de connecteur ECTP-5 non négatif en cas de succès ou `-1` dans les cas des causes d'erreur dont la liste est donnée dans le Tableau A.4.

**Tableau A.4 – Codes d'erreur associés à la fonction `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

### A.2.2 `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 de 32 bits ou d'une adresse IPv6 de 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, int role, struct ectp5_option
*options);
```

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;
- *role*: spécifie le rôle de l'initiateur appelant, par exemple TCN, LO ou LE;
- *options*: définit les options d'une connexion multidiffusée n-plex si *role* est défini par TCN.

**Tableau A.5 – Paramètre *role* de l'utilisateur du connecteur pour la fonction `mbind`**

role	Description
TCN	Créateur de la connexion et expéditeur en qualité de propriétaire dans les communications ECTP-5
LO	Destinataire prenant la responsabilité des transmissions dans une hiérarchie arborescente
LE	Destinataire qui n'est pas désigné comme propriétaire local (LO)

**Tableau A.6 – Champs du paramètre *options* de la fonction `mbind`**

Champs du paramètre options	Description
TCO	Option de configuration d'arborescence. '10' par défaut
AGN	Fréquence de création de paquets ACK. '32' par défaut
MSS	Taille maximale de segment. '1024' par défaut

Une application peut effectuer une fonction `mbind`, c'est-à-dire associer une adresse IP spécifique et une adresse de groupe de réseau au connecteur qui lui est associé. Les adresses de la source et du groupe doivent appartenir à une interface sur l'hôte.

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

**Tableau A.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 existant 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

### A.2.3 `maccept()`

Seuls l'entité feuille (LE) et le propriétaire local (LO) peuvent invoquer cette fonction. Ils peuvent en attendre l'activation par le propriétaire de la connexion TC (TCN) pendant un délai défini par le paramètre *timeout* et indiquer si la connexion multidiffusée a été établie ou non.

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

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 de la connexion TC);
- *raddrlen*: pointeur sur la taille de la structure d'adresse du connecteur pointé par *raddr*;
- *timeout*: valeur de temporisation (en secondes) concernant le délai d'attente du paquet CR émanant du propriétaire de la connexion TC.

Si l'application de la fonction `maccept` donne un résultat positif, elle renvoie la même valeur comme premier argument, *msockfd*. Après cela, la valeur renvoyée est appelée 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 dont la liste est donnée dans le Tableau A.8.

**Tableau A.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

**A.2.4 mconnect()**

La fonction `mconnect` est utilisée par le propriétaire de la connexion TC (TCN) ou par l'entité feuille (LE) qui est un participant 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 sur une adresse de groupe;
- *daddrlen*: taille de l'élément *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 A.9.

**Tableau A.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 rejetée de force
ENETUNREACH	Le réseau ne peut pas être atteint 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 propriétaire de la connexion TC a décliné la demande de participation de l'entité feuille (LE) ou du propriétaire local (LO)
ETIMEDOUT	La temporisation associée à l'établissement d'une connexion est expirée. Indique qu'il n'y a pas de réponse provenant du propriétaire de la connexion TC

**A.2.5 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 utilisé.

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 A.10.

Tableau A.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 <code>msockfd</code> 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

### A.2.6 `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'utilisateur expéditeur SU;
- *fromaddrlen*: taille de *fromaddr*.

Lorsqu'une application reçoit des données en provenance du tampon, elle peut identifier l'expéditeur (ou utilisateur expéditeur SU) 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 A.11.

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

Code d'erreur	Description
EBADF	L'argument <code>msockfd</code> 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 <code>msockfd</code> 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 propriétaire de la connexion TC a mis fin à la session
ETOEXPEL	Le propriétaire de la connexion TC a exclu l'entité feuille (LE) ou le propriétaire local (LO)

**A.2.7 mclose()**

La fonction `mclose` est utilisée pour quitter une connexion multidiffusée n-plex ou y mettre fin par fermeture du connecteur. L'action par défaut de `mclose` avec un connecteur ECTP-5 consiste à marquer le connecteur comme étant fermé et à revenir au processus immédiatement. Le descripteur de connecteur n'est plus utilisable par le processus.

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

Description des paramètres:

- *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 A.12.

**Tableau A.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

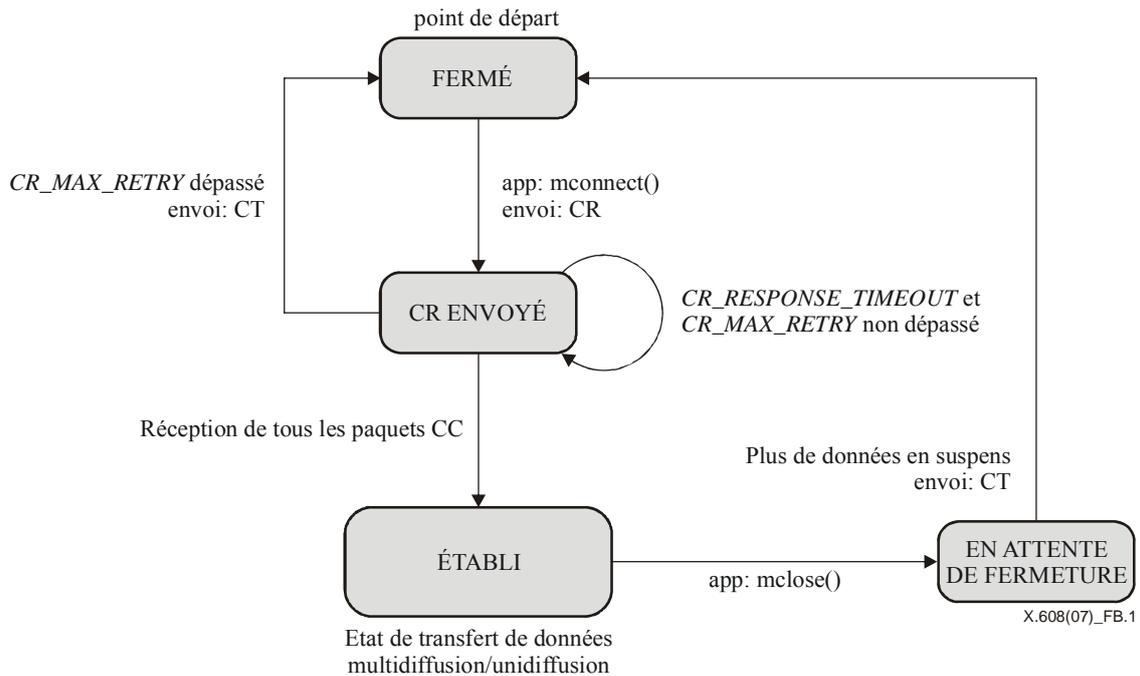
## Annexe B

## Diagrammes de transition d'état

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

La présente annexe indique les diagrammes de transition d'état des nœuds, du propriétaire de la connexion TC, du propriétaire local (LO) et de l'utilisateur du service TS dans le protocole ECTP-5 afin de faciliter l'implémentation de la présente Recommandation | Norme internationale.

La Figure B.1 montre le diagramme de transition d'état du propriétaire de la connexion TC dans le protocole ECTP-5, qui est décrit conformément au § 8.



**Figure B.1 – Diagramme de transition d'état du propriétaire de la connexion TC**

La Figure B.2 montre le diagramme de transition d'état du propriétaire local (LO) et de l'utilisateur du service TS dans le protocole ECTP-5, qui est décrit conformément au § 8.

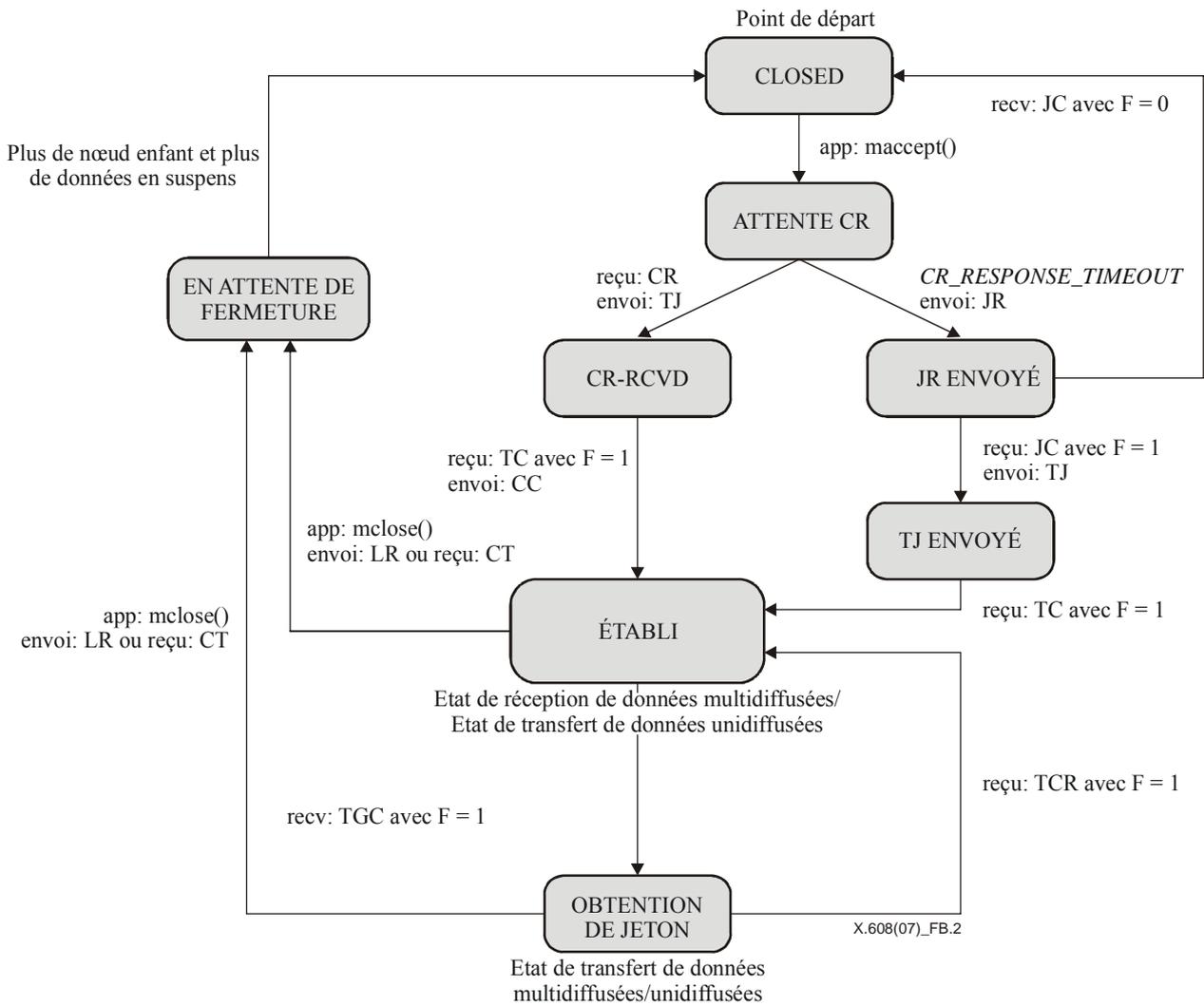


Figure B.2 – Diagramme de transition d'état du propriétaire local (LO) et de l'utilisateur du service TS dans le protocole ECTP-5

## Annexe C

## Exemple de valeurs de paramètres système dans le protocole ECTP-5

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

La présente annexe donne des exemples de valeurs de paramètres système dans le protocole ECTP-5 auxquels pourraient se référer les réalisateurs du protocole ECTP-5. Voir le Tableau C.1.

Tableau C.1 – Exemples de valeurs de paramètres système dans le protocole ECTP-5

Nom	Valeur par défaut
<i>ACK_GENERATION_NUM</i>	32
<i>CCR_MAX_RETRY</i>	5
<i>CCR_RETRY_TIMEOUT</i>	200 ms
<i>CR_MAX_RETRY</i>	5
<i>CR_RESPONSE_TIMEOUT</i>	5 s
<i>JR_MAX_RETRY</i>	5
<i>JR_RETRY_TIMEOUT</i>	200 ms
<i>MAX_SEGMENT_SIZE</i>	1024 octets
<i>NACK_MAX_RETRY</i>	5
<i>NACK_RETRY_TIMEOUT</i>	200 ms
<i>PB_MAX_RETRY</i>	5
<i>PB_PACKET_INT</i>	3 s
<i>PB_RETRY_TIMEOUT</i>	500 ms
<i>TCR_MAX_RETRY</i>	5
<i>TCR_RETRY_TIMEOUT</i>	200 ms
<i>TD_PACKET_INT</i>	5 ms
<i>TD_PACKET_NUM</i>	1000
<i>TD_PACKET_SIZE</i>	512 octets
<i>TDR_MAX_RETRY</i>	5
<i>TDR_RETRY_TIMEOUT</i>	200 ms
<i>TGR_MAX_RETRY</i>	5
<i>TGR_RETRY_TIMEOUT</i>	200 ms
<i>TJ_MAX_RETRY</i>	5
<i>TJ_RETRY_TIMEOUT</i>	200 ms
<i>TLR_MAX_RETRY</i>	5
<i>TLR_RETRY_TIMEOUT</i>	200 ms
<i>TNR_MAX_RETRY</i>	5
<i>TNR_RETRY_TIMEOUT</i>	200 ms
<i>TRR_MAX_RETRY</i>	5
<i>TRR_RETRY_TIMEOUT</i>	200 ms
<i>TSR_ARRIVAL_TIMEOUT</i>	15 s
<i>TSR_PACKET_INT</i>	5 s
<i>TSRR_MAX_RETRY</i>	5
<i>TSRR_RETRY_TIMEOUT</i>	500 ms

## **ISO/CEI 14476-5:2008 (F)**

Ces valeurs sont choisies pour l'environnement suivant:

- taille de session: 30 utilisateurs du service TS (30 utilisateurs expéditeurs SU);
- nombre de groupes locaux: 3 propriétaires locaux et 3 groupes locaux;
- débit d'envoi: 512 kbit/s;
- largeur de bande de liaison: 100 Mbit/s;
- retard de liaison: 40~50 ms entre groupes locaux et 10~25 ms dans un groupe local;
- taux d'erreur de bout en bout: 0,05~0,25.



## SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
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	Gestion des télécommunications y compris le RGT et maintenance des réseaux
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
<b>Série X</b>	<b>Réseaux de données, communication entre systèmes ouverts et sécurité</b>
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication