Groupe de travail Réseau
Request for Comments : 2022
Catégorie : En cours de normalisation

G. Armitage, Bellcore
novembre 1996
Traduction Claude Brière de L'Isle

Prise en charge de la diffusion groupée sur réseaux ATM fondés sur UNI 3.0/3.1

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

La transposition du service de diffusion groupée IP sans connexion en services ATM orientés connexion fournis par UNI 3.0/3.1 est une tâche qui n'est pas triviale. Le présent mémoire décrit un mécanisme pour prendre en charge les besoins de diffusion groupée des protocoles de couche 3 en général, et décrit son application à la diffusion groupée IP en particulier.

Les hôtes et routeurs IP fondés sur ATM utilisent un serveur de résolution d'adresse de diffusion groupée (MARS, *Multicast Address Resolution Server*) pour prendre en charge la diffusion groupée IP de niveau 2 du style décrit par la RFC1112 sur le service de connexion de point à multipoint de la norme UNI 3.0/3.1 de l'ATM Forum. Des grappes de points d'extrémité partagent un MARS et l'utilisent pour retracer et disséminer les informations qui identifient les nœuds répertoriés comme receveurs pour des groupes de diffusion groupée donnés. Cela permet aux points d'extrémité d'établir et de gérer les circuits virtuels *(VC)* en point à multipoint lors des transmissions au groupe.

Le comportement du MARS permet à la diffusion groupée de couche 3 d'être prise en charge en utilisant soit des maillages de VC soit des serveurs de diffusion groupée de niveau ATM. Ce choix peut être fait sur la base du groupe, et il est transparent pour les points d'extrémité.

Table des Matières

1. Introduction

La diffusion groupée est le processus par lequel un hôte de source ou une entité de protocole envoie simultanément un paquet à plusieurs destinations en utilisant une seule opération de "transmission" locale. Les cas les plus familiers d'envoi individuel et de diffusion peuvent être considérés comme des cas particuliers de diffusion groupée (où le paquet est livré, respectivement, à une seule destination, ou à "toutes" les destinations).

La plupart des modèles de couche réseau, comme celui décrit dans la RFC1112 [1] pour la diffusion groupée IP, supposent que les sources peuvent envoyer leurs paquets à des "adresses de diffusion groupées" abstraites. La prise en charge par la couche liaison d'une telle abstraction est supposée exister, et est fournie par des technologies telles que Ethernet.

ATM est utilisé comme nouvelle technologie de couche liaison pour prendre en charge divers protocoles, y compris IP. Avec la RFC1483 [2], l'IETF a défini un mécanisme multi protocoles pour encapsuler et transmettre des paquets en utilisant AAL5 sur des canaux virtuels (VC) ATM. Cependant, le Forum ATM a actuellement publié des spécifications de signalisation (UNI 3.0 [8] et UNI 3.1 [4]) qui ne fournissent pas l'abstraction d'adresse de diffusion groupée. Les connexions d'envoi individuel sont prises en charge par des VC bidirectionnels en point à point. La diffusion groupée est prise en charge par des VC unidirectionnels en point à multipoint. La limitation clé est que l'envoyeur doit avoir une connaissance préalable de chaque receveur prévu, et établir explicitement un VC avec lui-même comme nœud racine et les receveurs comme nœuds d'extrémité.

Le présent document a deux objectifs principaux :

- Définir un mécanisme d'enregistrement d'adresse de groupe et de répartition des adhésions qui permette à des réseaux fondés sur UNI 3.0/3.1 de prendre en charge le service de diffusion groupé de protocoles tels que IP.

- Définir des comportements spécifiques du point d'extrémité pour gérer les circuits virtuels en point à multipoint pour réaliser la diffusion groupée de paquets de couche 3.

Comme l'IETF est actuellement sur le point d'utiliser la diffusion groupée sur grande zone, les descriptions du présent document vont souvent se concentrer sur le modèle de service IP de la RFC1112. Un chapitre final parlera de l'application multi protocole de l'architecture.

Le présent document évite la discussion d'un aspect non trivial de l'utilisation de l'ATM - la spécification de la QS pour les VC établis en réponse à des besoins de couche supérieure. Les recherches dans ce domaine sont toujours très formatrices

[7], et on suppose donc que de futurs documents vont clarifier la transposition des exigences de QS en établissement de VC. La position par défaut à l'heure actuelle est que les VC sont établis avec une demande de service en débit binaire non spécifié (UBR, *Unspecified Bit Rate*) selon la typologie de l'IETF sur l'utilisation des VC pour IP en envoi individuel, décrit dans la RFC1755 [6].

1.1 Le serveur de résolution d'adresse de diffusion groupée

Le serveur de résolution d'adresse de diffusion groupée (MARS, *Multicast Address Resolution Server*) est une analogie étendue du serveur ARP ATM introduit par la RFC1577 [3]. Il agit comme un registraire, associant les identifiants de groupe de diffusion groupée de couche 3 aux interfaces ATM qui représentent les membres du groupe. Les messages MARS prennent en charge la distribution des informations sur les membres du groupe de diffusion groupée entre MARS et les points d'extrémité (hôtes ou routeurs). Les entités de résolution d'adresse de diffusion groupée interrogent le MARS lorsque une adresse de couche 3 doit être résolue pour l'ensemble des points d'extrémité ATM qui constituent le groupe à tout moment. Les points d'extrémité informent le MARS lorsque ils ont besoin de se joindre ou de quitter un groupe de couche 3 particulier. Pour fournir la notification asynchrone des changements d'adhésion au groupe, le MARS gère un circuit virtuel point à multipoint sur tous les points d'extrémité qui désirent prendre en charge la diffusion groupée.

Il existe des arguments valides en faveur des deux approches différentes de la diffusion groupe de niveau ATM de paquets de couche 3 – par un maillage de circuits virtuels en point à multipoint, ou par des serveurs de diffusion groupée (MCS, *multicast server*) de niveau ATM. L'architecture MARS permet d'utiliser aussi bien les maillages de circuits virtuels que les MCS groupe par groupe.

1.2 La grappe de diffusion groupée de niveau ATM

Chaque MARS gère une "grappe" de points d'extrémité rattachés à ATM. Une grappe est définie comme l'ensemble des interfaces ATM qui choisissent de participer à des connexions ATM directes pour réaliser la diffusion groupée de AAL_SDU entre elles-mêmes.

En pratique, une grappe est l'ensemble des points d'extrémité qui choisissent d'utiliser le même MARS pour enregistrer leurs adhésions et en recevoir leurs mises à jour.

Cette définition implique que le trafic entre les interfaces qui appartiennent à des grappes différentes passe par un appareil inter-grappes. (Dans le monde IP un appareil inter-grappes serait un routeur de diffusion groupée IP avec des interfaces logiques dans chaque grappe.) Le présent document évite explicitement de spécifier la nature des protocoles d'acheminement inter-grappe (de couche 3).

La transposition des grappes en d'autres ensembles obligés de points d'extrémité (comme des sous-réseaux logiques IP en envoi individuel) est laissée à chaque administrateur de réseau. Cependant, pour les besoins de la conformité au présent document, les administrateurs de réseau DOIVENT s'assurer que chaque sous-réseau IP logique (LIS, *Logical IP Subnet*) est desservi par un MARS distinct, créant une correspondance biunivoque entre la grappe et le LIS en envoi individuel. Les routeurs de diffusion groupée IP s'interconnectent alors à chaque LIS comme ils le font avec les sous-réseaux conventionnels. (L'assouplissement de cette restriction ne PEUT se faire qu'après l'achèvement des recherches en cours sur l'interaction entre les protocoles existants d'acheminement de diffusion groupée de couche 3 et les frontières de sous-réseau d'envoi individuel.)

Le terme de "membre de grappe" est utilisé dans le présent document pour se référer à un point d'extrémité qui utilise actuellement un MARS pour prendre en charge la diffusion groupée. La portée potentielle d'une grappe peut être l'ensemble des membres d'un LIS, alors que la portée réelle d'une grappe dépend des points d'extrémité qui sont réellement membres d'une grappe à un moment donné.

1.3 Présentation du document

Le présent document suppose la compréhension des concepts expliqués plus en détail dans la RFC1112 [1], la RFC1577 [3], UNI 3.0 [8]/3.1 [4], et la RFC1755 [6].

La Section 2 donne une vue d'ensemble de la diffusion groupée IP et de ce que la RFC1112 exige de Ethernet.

La Section 3 décrit plus en détails les services de prise en charge de la diffusion groupée offerts pae UNI 3.0/3.1, et souligne les différences entre les maillages de circuits virtuels et les serveurs de diffusion groupée (MCS) comme mécanismes de distribution de paquets à des destinations multiples.

La Section 4 donne une vue d'ensemble du MARS et de ses relations aux points d'extrémité ATM. Cette section traite aussi de l'encapsulation et de la structure des messages de contrôle MARS.

La Section 5 définit en substance le comportement du point d'extrémité membre de grappe, du côté receveur et du côté émetteur. Cela inclut le fonctionnement normal et la récupération d'erreur.

La Section 6 résume le comportement exigé d'un MARS.

La Section 7 regarde comment un serveur de diffusion groupée (MCS) interagit avec un MARS.

La Section 8 discute la façon dont les routeurs de diffusion groupée IP peuvent faire un nouvel usage des adhésions à des groupes disparates *(promiscuous)* et semi disparates. Un mécanisme conçu pour réduire la quantité de trafic IGMP produit par les routeurs est aussi exposé.

La Section 9 expose comment ce document s'applique dans le cas le plus général (non IP).

La Section 10 résume les propositions clés, et identifie les domaines de futures recherches qui sont générées par cette architecture de MARS.

Les appendices apportent des explications sur des questions qui découlent de la mise en œuvre du présent document. L'Appendice A expose les algorithmes de MARS et de point d'extrémité pour analyser les messages de MARS. L'Appendice B décrit les problèmes particuliers introduits par les paradigmes IGMP actuels, et les contournements intermédiaires possibles. L'Appendice C discute plus en détails du concept de "grappe", tandis que l'Appendice D expose brièvement un algorithme d'analyse des listes de TLV. L'Appendice E résume diverses valeurs de temporisateur utilisées dans le présent document, et l'Appendice F donne des exemples de pseudo-code pour une entité MARS.

1.4 Conventions.

Dans ce document, les règles suivantes de codage et de représentation de paquet sont utilisées :
- Tous les paramètres multi octets sont codés en forme gros boutienne (c'est-à-dire, l'octet de poids fort en premier).
- Dans tous les paramètres multi bits, la numérotation des bits commence à 0 pour le bit de moindre poids mémorisé (c'est-à-dire que le n^{ème}bit a le poids 2^n).
- Un bit établi a la valeur 1.
- Un bit qui est "ôté", '"non établi", ou à "zéro" a la valeur 0.

2. Résumé du modèle de service de diffusion groupée IP

Avec IP version 4 (IPv4), les adresses dans la gamme entre 224.0.0.0 et 239.255.255.255 (224.0.0.0/4) sont appelées adresses de "classe D" ou adresses de "diffusion groupée". Elles représentent de façon abstraite tous les hôtes IP de l'Internet (ou un sous-ensemble obligé de l'Internet) qui ont décidé de se "joindre" au groupe spécifié.

La RFC1112 exige qu'une interface IP capable de prendre en charge la diffusion groupée accepte la transmission de paquets IP à une adresse IP de groupe de diffusion groupée, que le nœud se considère ou non comme un "membre" de ce groupe. Par conséquent, l'adhésion au groupe est en effet sans pertinence pour le côté émetteur des interfaces de la couche liaison. Lorsque Ethernet est utilisé comme couche liaison (c'est l'exemple utilisé dans la RFC1112) aucune résolution d'adresse n'est requise pour transmettre les paquets. Une transposition algorithmique est effectuée localement de l'adresse de diffusion groupée IP en adresse de diffusion groupée Ethernet avant que le paquet ne soit envoyé de l'interface locale de la même manière "envoie et oublie" que pour un paquet IP en envoi individuel.

Se joindre et quitter un groupe de diffusion groupée IP est encore plus explicite du côté receveur – avec les primitives JoinLocalGroup et LeaveLocalGroup qui affectent les groupes dont l'interface locale de couche liaison devrait accepter les paquets. Lorsque la couche IP veut recevoir des paquets d'un groupe, elle produit une JoinLocalGroup. Quand elle ne veut plus recevoir de paquets, elle produit une LeaveLocalGroup. Un point clé à noter est que l'état qui change est une affaire locale, il n'a pas d'effet sur les autres hôtes rattachés à l'Ethernet.

IGMP est défini dans la RFC1112 comme prenant en charge les routeurs de diffusion groupée IP rattachés à un sous-réseau donné. Les hôtes produisent des messages de rapport IGMP lorsque ils effectuent une JoinLocalGroup, ou en réponse à un routeur de diffusion groupée IP qui envoie des interrogations IGMP. En transmettant périodiquement des interrogations, les routeurs de diffusion groupée IP sont capables d'identifier quels groupes de diffusion groupée IP ont une adhésion non à zéro sur un sous-réseau donné.

Une adresse de diffusion groupée IP spécifique, 224.0.0.1, est allouée à la transmission des messages d'interrogation IGMP. Les couches IP des hôtes produisent une JoinLocalGroup pour 224.0.0.1 lorsque elles entendent participer à la diffusion groupée IP, et produisent une LeaveLocalGroup pour 224.0.0.1 lorsque elles cessent de participer à la diffusion groupée IP.

Chaque hôte conserve une liste des groupes de diffusion groupée IP auxquels il a envoyé un JoinLocalGroup. Lorsque un routeur produit une interrogation IGMP sur 224.0.0.1, chaque hôte commence à envoyer des rapports IGMP pour chaque groupe dont il est membre. Les rapports IGMP sont envoyés à l'adresse du groupe, et non à 224.0.0.1, "afin que les autres membres du même groupe sur le même réseau puisse entendre le rapport" et ne se soucient pas d'en envoyer un de leur côté. Les routeurs de diffusion groupée IP concluent qu'un groupe n'a plus de membres sur le sous-réseau lorsque les interrogations IGMP ne suscitent plus de réponses associées.

3. Prise en charge de la diffusion groupée intra grappe par UNI 3.0/3.1

Pour les besoins du protocole MARS, UNI 3.0 et UNI 3.1 fournissent tous deux une prise en charge équivalente à la diffusion groupée. Les différences entre UNI 3.0 et UNI 3.1 sur les éléments de signalisation exigés sont traitées dans la RFC1755.

Le présent document décrira son fonctionnement en termes de fonctions "génériques" qui devraient être disponibles aux clients d'une entité de signalisation UNI 3.0/3.1 dans un point d'extrémité ATM donné. Le modèle ATM décrit un "utilisateur AAL" comme toute entité qui établit et gère des circuits virtuels et les services AAL sous-jacents pour échanger des données. Une interface IP sur ATM est une forme d'utilisateur AAL (bien que le mode d'encapsulation LLC/SNAP par défaut spécifié dans la RFC1755 exige réellement qu'une "entité LLC" soit l'utilisateur AAL qui à son tour prend en charge l'interface IP/ATM).

Les limitations les plus fondamentales de la prise en charge de la diffusion groupée par UNI 3.0/3.1 sont que :
- seuls des circuits virtuels point à multipoint, unidirectionnels peuvent être établis,
- seul le nœud racine (source) d'un circuit virtuel donné peut ajouter ou retirer des nœuds d'extrémité.

Les nœuds d'extrémité sont identifiés par leurs adresses ATM d'envoi individuel. UNI 3.0/3.1 définit deux formats d'adresse ATM – le E.164 natif et NSAP (bien qu'on doive souligner que l'adresse NSAP est ainsi appelée parce qu'elle utilise le format NSAP *(point d'accès au service réseau)* – un point d'extrémité ATM N'EST PAS un point de terminaison de couche réseau). En UNI 3.0/3.1 un "numéro ATM" est la principale identification d'un point d'extrémité ATM, et il peut utiliser l'un ou l'autre format. Dans certaines circonstances un point d'extrémité ATM doit être identifié à la fois par une adresse E.164 native (qui identifie le point de rattachement d'un réseau privé à un réseau public) et par une adresse NSAP ("sous-adresse ATM") qui identifie le point d'extrémité final au sein du réseau privé. Dans la suite de ce document, le terme sera utilisé pour signifier aussi bien un seul "numéro ATM" qu'un "numéro ATM" combiné avec une sous-adresse ATM.

3.1 Maillages de circuits virtuels

L'approche la plus fondamentale de la diffusion groupée intra grappe est le maillage de circuits virtuels (VC) de diffusion groupée. Chaque source établit son propre VC point à multipoint indépendant (une seule arborescence de diffusion groupée) pour l'ensemble des nœuds d'extrémité (destinations) dont il a été dit qu'ils sont membres du groupe auquel elle souhaite envoyer des paquets.

Les interfaces qui sont à la fois des envoyeurs et des membres du groupe (des nœuds d'extrémité) pour un groupe donné vont générer un VC en point à multipoint, et terminer un VC pour chaque autre envoyeur actif pour le groupe. Ce croisement de VC à travers le réseau ATM a donné le nom de "maillage de VC".

3.2 Serveurs de diffusion groupée

Un autre modèle fait que chaque source établit un VC avec un nœud intermédiaire – le serveur de diffusion groupée (MCS). Le serveur de diffusion groupée lui-même établit et gère un VC en point à multipoint vers les destinations désirées réelles.

Le MCS réassemble les AAL_SDU qui arrivent sur tous les VC entrants, et les met alors en file d'attente pour être transmis sur un seul VC en point à multipoint sortant. (Le réassemblage des AAL_SDU entrants est exigé au serveur de diffusion groupée car AAL5 ne prend pas en charge le multiplexage au niveau de la cellule des différents AAL_SDU sur un seul VC sortant.)

Les nœuds d'extrémité du VC en point à multipoint du serveur de diffusion groupée doivent être établis avant la transmission du paquet, et le serveur de diffusion groupée exige un mécanisme externe pour les identifier. Un effet colatéral de cette méthode est que les interfaces ATM qui sont à la fois des sources et des membres du groupe vont recevoir des copies de leurs propres paquets en retour du MCS. (Une méthode de remplacement est que le serveur de diffusion groupée retransmette explicitement les paquets sur des VC individuels entre eux-mêmes et les membres du groupe. Un avantage de cette seconde approche est que le serveur de diffusion groupée peut s'assurer que les sources ne reçoivent pas de copie de leurs propres paquets.)

Le plus simple MCS ne prête aucune attention au contenu de chaque AAL_SDU. Il est un pur appareil de niveau AAL/ATM. Des architectures de MCS plus complexes (où un seul point d'extrémité dessert plusieurs groupes de couche 3) sont possibles, mais sortent du domaine d'application de ce document. Un exposé plus détaillé figure à la Section 7.

3.3 Compromis

Les disputes sur les mérites respectifs des maillages de circuits virtuels et des serveurs de diffusion groupée ont fait rage pendant un certain temps. Le choix ultime dépend des compromis qu'un administrateur de système doit faire entre le débit, la latence et la consommation de ressources. Même des critères tels que la latence peuvent signifier des choses différentes pour des personnes différentes – c'est le temps que met le paquet de bout en bout, ou le temps qu'il faut pour établir un groupe après un changement dans les adhérents ? Le choix final dépend des caractéristiques des applications qui génèrent le trafic en diffusion groupée.

Si on se concentre sur le chemin des données, on peut préférer le maillage de VC à cause de l'absence de l'évident goulot d'étranglement obligé d'un MCS. Le débit sera vraisemblablement supérieur, et la latence de bout en bout inférieure parce que le maillage n'a pas le réassemblage intermédiaire d'AAL_SDU qui doit survenir dans les MCS. Le système de signalisation ATM sous-jacent a aussi une plus grande opportunité d'assurer des points d'embranchement optimaux aux commutateurs ATM le long de l'arborescence de diffusion groupée générée sur chaque source.

Cependant, la consommation de ressources sera plus élevée. Chaque interface ATM de membre du groupe doit terminer un VC par envoyeur (ce qui consomme de la mémoire embarquée pour les informations d'état, l'instance de service AAL, et la mise en mémoire tampon conformément à l'architecture particulière du fabricant). Au contraire, avec un serveur de diffusion groupée, seuls deux circuits virtuels (un en sortie, un en entrée) sont nécessaires, indépendamment du nombre d'envoyeurs. L'allocation de ressources en rapport avec les VC est aussi inférieure au sein du nuage ATM lorsque on utilise un serveur de diffusion groupée. Ces points peuvent être considérés comme décisifs dans des environnements où les VC, à travers la UNI *(interface usager réseau)* ou au sein du nuage ATM, sont coûteux (par exemple, la facturation ATM fondée sur le nombre de circuits virtuels), ou dans les contextes d'AAL, sont limités dans les interfaces ATM de points d'extrémité.

Si on se concentre sur la charge de signalisation, les MCS ont alors l'avantage lorsque on est confronté à des ensembles dynamiques de receveurs. Chaque fois que changent les adhérents à un groupe de diffusion groupée (un nœud d'extrémité doit être ajouté ou retiré) un seul VC en point à multipoint doit être modifié lorsque on utilise un MCS. Cela ne génère qu'un seul événement de signalisation à travers l'UNI du MCS. Cependant, lorsque le changement d'adhésion survient dans un maillage de VC, les événements de signalisation surviennent aux UNI de chaque source de trafic – la charge de signalisation transitoire est proportionnelle au nombre de sources. Cela a des conséquences évidentes si on définit la latence comme le délai de stabilisation de la connexité d'un groupe après un changement (en particulier lorsque le nombre d'envoyeurs augmente).

Finallement, comme on l'a noté plus haut, les MCS introduisent un problème de "paquet réfléchi", qui exige que des informations d'AAL_SDU supplémentaires soient portées afin que les sources de couche 3 détectent le retour de leurs propres AAL_SDU.

L'architecture MARS permet aux administrateurs de système d'utiliser l'une ou l'autre approche groupe par groupe.

3.4 Interaction avec l'entité de signalisation UNI 3.0/3.1 locale

Les fonctions génériques de signalisation suivantes sont présumées disponibles aux usagers AAL locaux :

L_CALL_RQ	Établir un VC en envoi individuel sur un point d'extrémité spécifique.
L_MULTI_RQ	Établir un VC en diffusion groupée pour un point d'extrémité spécifique
L_MULTI_ADD	Ajouter un nouveau nœud d'extrémité au VC précédemment établi
L_MULTI_DROP	Retirer un nœud d'extrémité spécifique du VC établi
L_RELEASE	Libérer le VC en envoi individuel, ou toutes les extrémités d'un VC en diffusion groupée

Les échanges de signalisation et les informations locales passées entre l'utilisateur AAL et l'entité UNI 3.0/3.1 de signalisation avec ces fonctions sortent du domaine d'application du présent document.

Les indications suivantes sont supposées être disponibles aux utilisateurs AAL, générées par l'entitée locale de signalisation UNI 3.0/3.1 :

L_ACK	Achèvement réussi d'une demande locale
L_REMOTE_CALL	Un nouveau VC a été établi avec l'utilisateur AAL
ERR_L_RQFAILED	Un point d'extrémité ATM distant a rejeté une L_CALL_RQ, L_MULTI_RQ, ou L_MULTI_ADD
ERR_L_DROP	Un point d'extrémité ATM distant a fermé un VC existant
ERR_L_RELEASE	Un VC existant s'est terminé.

Les échanges de signalisation et les informations locales passées entre utilisateur AAL et entité de signalisation UNI 3.0/3.1 avec ces fonctions sortent du domaine d'application du présent document.

4. Vue d'ensemble du MARS

Le MARS peut résider au sein de tout point d'extrémité ATM directement adressable par les points d'extrémité qu'il dessert. Les points d'extrémité qui souhaitent se joindre à une grappe de diffusion groupée doivent être configurés avec l'adresse ATM du nœud sur lequel réside le MARS de la grappe. (Le paragraphe 5.4 décrit comment des MARS de sauvegarde peuvent être ajoutés pour prendre en charge les activités d'une grappe. Les références au "MARS" dans la suite du texte sont supposées signifier le MARS actif pour la grappe.)

4.1 Architecture

Du point de vue architectural, le MARS est une évolution du serveur ARP de la RFC1577. Alors que le serveur ARP conserve un tableau des paires d'adresses {IP, ATM} pour tous les points d'extrémité IP dans un LIS, le MARS conserve des tableaux extensifs des transpositions {adresse de couche 3, ATM.1, ATM.2, ... ATM.n}. Il peut être configuré avec certaines transpositions, ou apprendre les transpositions de façon dynamique. Le format du champ {adresse de couche 3} n'est généralement pas interprété par le MARS.

Un seul nœud ATM peut prendre en charge plusieurs MARS logiques, chacun d'eux prenant en charge une grappe distincte. La contrainte est que chaque MARS ait une unique adresse ATM (par exemple, un champ SEL différent dans l'adresse NSAP du nœud sur lequel résident les divers MARS). Par définition, une seule instance d'un MARS ne peut pas prendre en charge plus d'une grappe.

Le MARS distribue les informations de mise à jour des adhésions au groupe des membres de la grappe sur un circuit virtuel en point à multipoint appelé le ClusterControlVC *(circuit virtuel de contrôle de grappe)*. De plus, lorsque les serveurs de diffusion groupée (les MCS) sont utilisés, il établit aussi un circuit virtuel point à multipoint distinct vers les MCS enregistrés, appelés le ServerControlVC *(circuit virtuel de contrôle de serveur)*. Tous les membres d'une grappe sont des nœuds d'extrémité de ClusterControlVC. Tous les serveurs de diffusion groupée enregistrés sont des nœuds d'extrémité de ServerControlVC (qui est décrit plus en détails à la Section 6).

Le MARS NE PARTICIPE PAS à la diffusion groupée réelle des paquets de données de couche 3.

4.2 Format du message de contrôle

Par défaut, tous les messages de contrôle MARS DOIVENT être encapsulés dans LLC/SNAP en utilisant les codets suivants :

[0xAA-AA-03][0x00-00-5E][0x00-03][message de contrôle MARS] (LLC) (OUI) (PID)

(C'est un PID tiré du OUI de l'IANA.)

Les messages de contrôle MARS sont constitués de quatre composants majeurs :

[En-tête fixe][Champs obligatoires][Adresses][TLV supplémentaires]

[En-tête fixe] contient des champs qui indiquent l'opération qui est effectuée et le protocole de couche 3 auquel on se réfère (par exemple, IPv4, IPv6, AppleTalk, etc). L'en-tête fixe porte aussi des informations de somme de contrôle, et des "trucs" pour permettre que cette structure de base de message de contrôle soit réutilisée par les autres protocoles d'interrogation/réponse.

La section [Champs obligatoires] porte des paramètres de longueur fixe qui dépendent du type d'opération indiqué dans [En-tête fixe].

La zone suivante [Adresses] porte des champs de longueur variable pour les adresses de source et de cible – à la fois de matériel (par exemple, ATM) et de couche 3 (par exemple, IPv4). Elles fournissent les informations fondamentales qu'utilisent les enregistrements, les interrogations, et les mises à jour, sur lesquelles elles opèrent. Car les champs du protocole MARS dans [En-tête fixe] indiquent comment interpréter le contenu de [Adresses].

[TLV supplémentaires] représente une liste facultative d'éléments d'informations de (type, longueur, valeur) codés qui peuvent être ajoutés pour fournir des informations supplémentaires. Ce dispositif est décrit en détail à la Section 10.

Les messages MARS contiennent des champs d'adresse de longueur variable. Dans tous les cas, les adresses nulles DEVRONT être codées avec la longueur zéro, et n'avoir pas d'espace alloué dans le message.

(L'encapsulation LLC/SNAP unique des messages de contrôle MARS signifie que la fonction de serveur MARS et ARP peut être mise en œuvre au sein d'une entité commune, et partager un circuit virtuel client-serveur, si la mise en œuvre en décide ainsi. Noter que le codet LLC/SNAP pour MARS est différent du codet utilisé pour ATMARP.)

4.3 Champs d'en-tête fixes dans les messages de contrôle MARS

La section [En-tête fixe] a le format suivant

Données :

mar$afn	16 bits : Famille d'adresses (0x000F).
mar$pro	56 bits : Identification du protocole
mar$hdrrsv	24 bits : Réservé. Non utilisé par le protocole de contrôle MARS
mar$chksum	16 bits : Somme de contrôle sur la totalité du message MARS
mar$extoff	16 bits : Décalage des extensions
mar$op	16 bits : Code d'opératon
mar$shtl	8 bits Type et longueur du numéro de source ATM. (r)
mar$sstl	8 bits : Type et longueur de sous-adresse de source ATM. (q)

mar$shtl et mar$sstl donnent des informations concernant l'adresse du matériel (ATM) de la source. Dans le protocole MARS, ces champs sont toujours présents, car chaque message MARS porte une adresse ATM de source non nulle. Dans tous les cas, l'adresse ATM de source est le premier champ de longueur variable dans la section [Adresses].

Les autres champs dans [En-tête fixe] sont décrits dans les paragraphes suivants.

4.3.1 Type de matériel

mar$afn définit le type d'adresse de couche liaison portée. La valeur de 0x000F DEVRA être utilisée par les messages MARS générés en conformité avec le présent document. Le codage des adresses et sous-adresses ATM lorsque mar$afn = 0x000F est décrit au paragraphe 5.1.2. Le codage lorsque mar$afn != 0x000F sort du domaine d'application du document.

4.3.2 Type de protocole

Le champ mar$pro est constitué de deux sous-champs :

mar$pro.type	16 bits	Type de protocole.
mar$pro.snap	40 bits	Extension facultative SNAP au type de protocole.

Le champ mar$pro.type est un entier non signé de 16 bits qui représente l'espace de nombre suivant :

0x0000 to 0x00FF	Protocoles définis par les NLPID équivalents.
0x0100 to 0x03FF	Réservé pour utilisation future par l'IETF.
0x0400 to 0x04FF	Alloué pour une utilisation par ATM Forum.

0x0500 to 0x05FF	Utilisation expérimentale/locale.
0x0600 to 0xFFFF	Protocoles définis par les Ethertypes équivalents.

(fondé sur l'observation que les Ethertypes valides ne sont jamais inférieurs à 0x600, et les NLPID jamais supérieurs à 0xFF.)

La valeur de NLPID de 0x80 est utilisée pour indiquer qu'une extension codée en SNAP est utilisée pour coder le type de protocole. Lorsque mar$pro.type == 0x80, l'extension SNAP est codée dans le champ mar$pro.snap. C'est appelé l'identifiant de protocole de "forme longue".

Si mar$pro.type != 0x80, le champ mar$pro.snap DOIT alors être à zéro en émission et ignoré en réception. Le champ mar$pro.type lui-même identifie le protocole auquel on se réfère. C'est ce qu'on appelle la "forme courte" d'identifiant de protocole.

Dans tous les cas, lorsque un protocole a un numéro alloué dans l'espace mar$pro.type (excluant 0x80) la forme courte DOIT être utilisée lors de la transmission de messages MARS. De plus, lorsque un protocole a des formes d'identification courtes et longues valides, les receveurs PEUVENT choisir de reconnaître la forme longue.

Les valeurs de mar$pro.type autres que 0x80 PEUVENT avoir la "forme longue" définie dans des documents futurs.

Pour la suite de ce document, les références à mar$pro DEVRONT être interprétées comme signifiant mar$pro.type, ou mar$pro.type en combinaison avec mar$pro.snap selon le cas approprié.

L'utilisation des différents types de protocoles est décrite en détails à la section 9.

4.3.3 Somme de contrôle

Le champ mar$chksum porte une somme de contrôle IP standard calculée sur la totalité du message de contrôle MARS (à l'exclusion de l'en-tête LLC/SNAP). Le champ est réglé à zéro avant d'effectuer le calcul de la somme de contrôle.

Comme le message MARS entier encapsulé dans LLC/SNAP est protégé par le CRC de 32 bits du transport AAL5, les mises en œuvre PEUVENT choisir d'ignorer le dispositif de somme de contrôle. Si aucune somme de contrôle n'est calculée, ces bits DOIVENT être remis à zéro avant transmission. Si aucune somme de contrôle n'est effectuée à réception, ce champ DOIT être ignoré. Si un receveur est capable de valider une somme de contrôle, il DOIT seulement effectuer la validation lorsque le champ mar$chksum reçu est différent de zéro. Les messages qui arrivent avec un mar$chksum de 0 sont toujours considérés comme valides.

4.3.4 Décalage d'extensions

Le champ mar$extoff identifie l'existence et la localisation d'une liste de paramètres facultatifs supplémentaires. Son utilisation est décrite à la Section 10.

4.3.5 Code d'opération

Le champ mar$op est subdivisé en deux champs de 8 bits – mar$op.version (octet de poids fort) et mar$op.type (octet final). Ensemble, ils indiquent la nature du message de contrôle, et le contexte au sein duquel [Champs obligatoires], [Adresses], et [TLV supplémentaires] devraient être interprétés.

mar$op.version

0	Protocole MARS défini dans ce document.
0x01 - 0xEF	Réservé pour utilisation future par l'IETF.
0xF0 - 0xFE	Alloué pour l'usage de l'ATM Forum.
0xFF	Utilisation expérimentale/locale.

mar$op.type
Sa valeur indique l'opération effectuée, au sein du contexte de la version du protocole de contrôle indiquée par mar$op.version.

Pour le reste de ce document, les références à la valeur de mar$op DEVRONT être comprises comme signifiant mar$op.type, avec mar$op.version = 0x00. Les valeurs utilisées dans ce document sont résumées à la Section 11.

(Noter que cet espace numérique est indépendant de l'espace numérique du code d'opération ATMARP.)

4.3.6 Réservé

mar$hdrrsv peut être subdivisé et recevoir des significations spécifiques pour d'autres protocoles de contrôle indiqués par mar$op.version != 0.

5. Comportement d'interface de point d'extrémité (client MARS)

Un point d'extrémité peut être vu comme une "cale" ou couche de "convergence", se tenant entre une interface de couche liaison d'un protocole de couche 3 et le service UNI 3.0/3.1 sous-jacent. Un point d'extrémité dans ce contexte peut exister dans un hôte ou un routeur – toute entité qui exige une interface génerique de "couche 3 sur ATM" pour la prise en charge de la diffusion groupée de couche 3. La présente section se découpe en deux sous sections clés – une pour le côté émission, et une pour le côté réception.

Plusieurs interfaces ATM logiques peuvent être prises en charge par une seule interface ATM physique (par exemple, en utilisant des valeurs de SEL différentes dans l'adresse formatée NSAP allouée à l'interface ATM physique). Donc ; les mises en œuvre DOIVENT permettre que plusieurs interfaces indépendantes de "couche 3 sur ATM", chacune avec son propre MARS configuré (ou tableau de MARS, comme expliqué au paragraphe 5.4) et la capacité à être rattachée à la même grappe ou à des grappes différentes.

Le chemin de signalisation initial entre un client MARS (qui gère un point d'extrémité) et son MARS associé est un VC transitoire bidirectionnel en point à point. Ce VC est établi par le client MARS, et est utilisé pour envoyer des interrogations au MARS et en recevoir des réponses. Il a un temporisateur d'inactivité associé, et il est démonté si il n'est pas utilisé pendant une durée configurable. La valeur minimum suggérée pour cette durée est de 1 minute, et la valeur RECOMMANDÉE par défaut est 20 minutes. (Lorsque le MARS et le serveur ARP sont co-résidents, ce VC peut être utilisé à la fois pour le trafic ATM de l'ARP et pour le trafic de contrôle du MARS.)

Le chemin de signalisation restant est ClusterControlVC, auquel le client MARS est ajouté comme nœud d'extrémité lorsque il s'enregistre (comme décrit au paragraphe 5.2.3).

La plus grande partie du présent document traite de la distribution des informations qui permettent aux points d'extrémité d'établir et gérer les VC point à multipoint sortants – les chemins de transmission pour le trafic de diffusion groupée vers des groupes particuliers de diffusion groupée. Le format réel des AAL_SDU envoyés sur ces VC est presque complètement en dehors du domaine d'application de la présente spécification. Cependant, les points d'extrémité ne sont pas supposés savoir si leur chemin de transmission conduit directement à des membres d'un groupe de diffusion groupée ou à un MCS (décrit à la section 3). Cela exige une encapsulation paquet par paquet supplémentaire (décrite au paragraphe 5.5) pour aider à la détection des AAL_SDU réfléchis.

5.1 Comportement du côté émission

La description qui suit va souvent se référer à une interface IPv4/ATM capable de transmettre des paquets à une adresse de classe D à tout instant, sans avertissement préalable. Il devrait être trivial à une mise en œuvre de généraliser ce comportement aux exigences d'un autre protocole de données de couche 3.

Lorsque une entité locale de couche 3 passe un paquet à transmettre, le point d'extrémité s'assure d'abord qu'il existe déjà un chemin de sortie vers le groupe de diffusion groupée de destination. Si ce n'est pas le cas, le MARS est interrogé sur un ensemble de points d'extrémité ATM qui représentent un chemin de transmission approprié. (Les points d'extrémité ATM peuvent représenter les membres du groupe réels au sein de la grappe, ou un ensemble d'un ou plusieurs MCS. Le point d'extrémité ne fait pas la distinction entre l'un et l'autre cas. Le paragraphe 6.2 décrit le comportement du MARS qui conduit à ce que les MCS soient fournis comme chemin de transmission pour un groupe de diffusion groupée.)

L'interrogation est exécutée en produisant une MARS_REQUEST. La réponse du MARS peut prendre une des deux formes :

MARS_MULTI Séquence de messages MARS_MULTI qui retourne l'ensemble des points d'extrémité ATM qui sont les nœuds d'extrémité d'un VC point à multipoint sortant (le chemin de transmission).

MARS_NAK Aucune transposition n'est trouvée, le groupe est vide.

Les formats de ces messages sont décrits au paragraphe 5.1.2.

Les VC sortants sont établis avec une demande de service à débit binaire non spécifié (UBR, *Unspecified Bit Rate*), selon la typologie de l'usage des VC de l'IETF pour IP en envoi individuel, décrite dans la RFC1755 [6]. De futurs documents peuvent s'écarter de cette approche et permettre la spécification de différents paramètres de trafic ATM à partir d'informations ou paramètres configurés en local obtenus par des moyens externes.

5.1.1 Restitution de l'appartenance au groupe par le MARS

Si le MARS n'a pas de transposition pour l'adresse de classe D désirée, un MARS_NAK sera retourné. Dans ce cas, le paquet IP DOIT être éliminé en silence. Si une correspondance est trouvée dans les tableaux du MARS, il va procéder à l'envoi en retour des adresses ATM.1 à ATM.n dans une séquence de un ou plusieurs MARS_MULTI. Un mécanisme simple est utilisé pour détecter et se récupérer des pertes de messages MARS_MULTI.

(Si le client apprend qu'il n'y a pas d'autre membre du groupe dans la grappe – le MARS retourne un MARS_NAK ou un MARS_MULTI au client qui est seul membre – il DOIT retarder l'envoi d'une nouvelle MARS_REQUEST pour ce groupe pendant une période non inférieure à 5 secondes et pas supérieure à 10 secondes.)

Chaque MARS_MULTI porte un champ booléen x, et un champ d'entier de 15 bits y – exprimé par MARS_MULTI(x,y). Le champ y agit comme un numéro de séquence, commençant à 1 et s'incrémentant à chaque MARS_MULTI envoyé. Le champ x agit comme un marqueur de "fin de réponse". Lorsque x == 1 la réponse du MARS est considérée comme terminée.

De plus, chaque MARS_MULTI peut porter plusieurs adresses ATM tirées de l'ensemble {ATM.1, ATM.2, ... ATM.n}. Un MARS DOIT minimiser le nombre de MARS_MULTI transmis en plaçant autant d'adresses de membres du groupe que possible dans un seul MARS_MULTI. La limite sur la longueur d'un message MARS_MULTI individuel DOIT être la MTU du circuit virtuel sous-jacent.

Par exemple, supposons que n adresses ATM doivent être retournées, chaque MARS_MULTI étant limité à seulement p adresses ATM, et p << n. Cela exigerait une séquence de k messages MARS_MULTI (où k = (n/p)+1, en utilisant une arithmétique d'entiers) transmise comme suit :

```
MARS_MULTI(0,1) rapporte {ATM.1 ... ATM.p}
MARS_MULTI(0,2) rapporte {ATM.(p+1) ... ATM.(2p)}
        [.......]
MARS_MULTI(1,k) rapporte { ... ATM.n}
```

Si k == 1 alors seul MARS_MULTI(1,1) est envoyé.

Les défaillances typiques sont la perte d'un ou plusieurs paquets de MARS_MULTI(0,1) à MARS_MULTI(0,k-1). Cette perte est détectée lorsque y saute de plus de un entre des MARS_MULTI consécutifs. Un autre mode de défaillance est la perte de MARS_MULTI(1,k). Un temporisateur DOIT être mis en œuvre pour signaler la défaillance du derneir MARS_MULTI à arriver. Une valeur par défaut de 10 secondes est RECOMMANDÉE.

Si un "saut de séquence" est détecté, l'hôte DOIT attendre pendant MARS_MULTI(1,k), éliminer tous les résultats, et répéter la MARS_REQUEST.

Si une fin de temporisation survient, l'hôte DOIT éliminer tous les résultats, et répéter la MARS_REQUEST.

Un dernier mode de défaillance implique le numéro de séquence MARS (décrit au paragraphe 5.1.4.2 et porté dans chaque partie d'un MARS_MULTI multi-parties). Si sa valeur change durant la réception d'un MARS_MULTI multi-parties l'hôte DOIT attendre MARS_MULTI(1,k), éliminer tous les résultats, et répéter la MARS_REQUEST.

(La corruption des contenus de cellules va conduire à la perte de MARS_MULTI par la défaillance du réassemblage AAL5 CPCS_PDU, qui sera détectée par les mécanismes décrits ci-dessus.

Si le MARS gère une grappe de points d'extrémité épars sur des réseaux ATM différents mais directement accessibles, il ne sera pas capable de retourner tous les membres du groupe dans un seul MARS_MULTI. Le format du message MARS_MULTI permet de retourner de l'E.164, du NSAP ISO, ou du (E.164 + NSAP) comme adresses ATM. Cependant, chaque message MARS_MULTI ne peut retourner des adresses ATM que de mêmes type et longueur. Les adresses retournées DOIVENT être groupées conformément au type (E.164, NSAP ISO, ou les deux) et retournées dans une séquence de parties MARS_MULTI séparées.

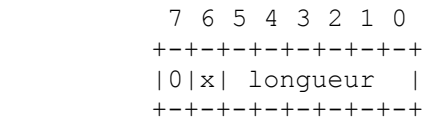
5.1.2 Messages MARS_REQUEST, MARS_MULTI, et MARS_NAK

Le message MARS_REQUEST est montré ci après. Il est indiqué par une "'valeur de type d'opération" (mar$op) de 1.

L'adresse de diffusion groupée à résoudre est placée dans le champ Adresse de protocole cible (mar$tpa) et l'adresse de matériel cible est réglée à nul (mar$thtl et mar$tstl sont tous deux à zéro). Dans les environnements IPv4, le type de protocole (mar$pro) est 0x800 et la longueur de l'adresse du protocole cible (mar$tpln) DOIT être réglée à 4. Les champs de source DOIVENT contenir le numéro ATM et la sous-adresse du client qui produit la MARS_REQUEST (la sous-adresse PEUT être nulle).

Données :		
mar$afn	16 bits	Famille d'adresse (0x000F).
mar$pro	56 bits	Identification du protocole
mar$hdrrsv	24 bits	Réservé. Non utilisé par le protocole de contrôle MARS
mar$chksum	16 bits	Somme de contrôle sur le message MARS entier
mar$extoff	16 bits	Décalage d'extensions
mar$op	16 bits	Code d'opération (MARS_REQUEST = 1)
mar$shtl	8 bits	Type & longueur du numéro ATM de source. (r)
mar$sstl	8 bits	Type & longueur de la sous-adresse ATM de source. (q)
mar$spln	8 bits	Longueur de l'adresse de protocole de source. (s)
mar$thtl	8 bits	Type & longueur du numéro ATM cible. (x)
mar$tstl	8 bits	Type & longueur de la sous-adresse ATM cible. (y)
mar$tpln	8 bits	Longueur de l'adresse de groupe cible. (z)
mar$pad	64 bits	Bourrage (aligne mar$sha sur MARS_MULTI).
mar$sha	roctets	Numéro ATM de source
mar$ssa	qoctets	Sous-adresse ATM de source.
mar$spa	soctets	Adresse du protocole de source.
mar$tpa	zoctets	Adresse de groupe de diffusion groupée de source
mar$tha	xoctets	Numéro ATM cible.
mar$tsa	yoctets	Sous-adresse ATM cible.

Suivant l'approche de la RFC1577, les champs mar$shtl, mar$sstl, mar$thtl et mar$tstl sont codés comme suit :



Le bit de poids fort est réservé et DOIT être mis à zéro. Le second bit de poids fort (x) est un fanion qui indique si l'adresse ATM à laquelle on se réfère est en format NSAPA de l'ATM Forum (x = 0), ou en format E.164 natif (x = 1).

Les 6 bits du fond sont une valeur d'entier non signé qui indique la longueur de l'adresse ATM associée en octets. Si cette valeur est zéro, le fanion x est ignoré.

Les champs mar$spln et mar$tpln sont des entiers non signés de 8 bits, qui donnent la longueur en octets des champs d'adresse de protocole, respectivement de source et de cible.

Les paquets MARS utilisent de vrais champs de longueur variable. Une adresse nulle (non existante) DOIT être codée avec la longueur zéro, et aucun espace n'est alloué pour elle dans le corps du message.

MARS_NAK est la MARS_REQUEST retournée avec la valeur de type d'opération de 6. Tous les autres champs sont laissés comme dans la MARS_REQUEST (par exemple, on ne transpose pas les informations de source et de cible. Dans tous les cas, les clients MARS utilisent les champs d'adresse de source pour identifier le retour de leurs propres messages).

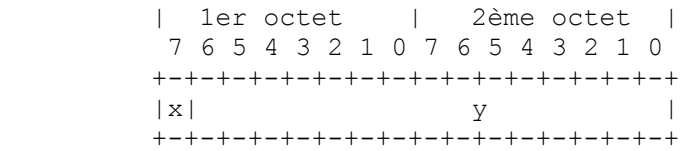
Le message MARS_MULTI est identifié par une valeur mar$op de 2. Le format du message est :

Données :		
mar$afn	16 bits	Famille d'adresse (0x000F).
mar$pro	56 bits	Identification du protocole
mar$hdrrsv	24 bits	Réservé. Non utilisé par le protocole de contrôle MARS.
mar$chksum	16 bits	Somme de contrôle sur le message MARS entier.

mar$extoff	16 bits	Décalage d'extensions.
mar$op	16 bits	Code d'opération (MARS_MULTI = 2).
mar$shtl	8 bits	Type & longueur du numéro de source ATM. (r)
mar$sstl	8 bits	Type & longueur de la sous-adresse de source ATM. (q)
mar$spln	8 bits	Longueur de l'adresse du protocole de source. (s)
mar$thtl	8 bits	Type & longueur du numéro ATM cible. (x)
mar$tstl	8 bits	Type & longueur de la sous-adresse ATM cible. (y)
mar$tpln	8 bits	Longueur de l'adresse de groupe cible. (z)
mar$tnum	16 bits	Nombre d'adresses cible ATM retournées. (N)
mar$seqxy	16 bits	Fanion booléen x et numéro de séquence y.
mar$msn	32 bits	Numéro de séquence MARS.
mar$sha	roctets	Numéro de source ATM.
mar$ssa	qoctets	Sous-adresse de source ATM.
mar$spa	soctets	Adresse du protocole de source.
mar$tpa	zoctets	Adresse cible du groupe de diffusion groupée.
mar$tha.1	xoctets	Cible ATM numéro 1.
mar$tsa.1	yoctets	Sous-adresse ATM cible 1
mar$tha.2	xoctets	Numéro ATM cible 2
mar$tsa.2	yoctets	Sous-adresse ATM cible 2
	[.....]	
mar$tha.N	xoctets	Numéro ATM cible N
mar$tsa.N	yoctets	Sous-adresse ATM cible N

Les champs protocole de source et adresse ATM sont copiés directement de la MARS_REQUEST à laquelle ce MARS_MULTI répond (et non le MARS lui-même).

mar$seqxy est codé avec le fanion x dans le bit de tête, et le numéro de séquence y codé comme un entier non signé dans les 15 bits restants.



mar$tnum est un entier non signé qui indique combien de paires de {martha,martsa} (c'est-à-dire, combien d'adresses ATM de membres du groupe) sont présentes dans le message. mar$msn est un nombre non signé de 32 bits rempli par le MARS avant de transmettre chaque MARS_MULTI. Son utilisation est décrite au paragraphe 5.1.4.

Par exemple, supposons que nous ayons une grappe de diffusion groupée qui utilise des adresses de protocole de 4 octets, des numéros ATM de 20 octets, et des sous-adresses ATM de 0 octet. Pour n membres du groupe dans un seul MARS_MULTI on a besoin d'un message de (60 + 20n) octets. Si on suppose que la MTU par défaut est de 9180 octets, on peut retourner un maximum de 456 adresses de membres du groupe dans un seul MARS_MULTI.

5.1.3 Établissement du circuit virtuel multipoint sortant

À la suite de l'achèvement de la réponse MARS_MULTI, le point d'extrémité peut établir un nouveau VC point à multipoint, ou en réutiliser un existant.

Si il établit un nouveau VC, une L_MULTI_RQ est produite pour ATM.1, suivie par une L_MULTI_ADD pour chaque membre de l'ensemble {ATM.2, ...ATM.n} (en supposant que l'ensemble est non nul). Le paquet est alors transmis sur le VC nouvellement créé tout comme ce serait fait pour un VC en envoi individuel.

Après avoir transmis le paquet, l'interface locale garde le VC ouvert et le marque comme chemin actif de sortie de l'hôte pour tout paquet IP ultérieur à envoyer à cette adresse de classe D.

Lors de l'établissement d'un nouveau VC de diffusion groupée, il est possible qu'échouent un ou plusieurs L_MULTI_RQ ou L_MULTI_ADD. La cause de la défaillance de UNI 3.0/3.1 doit être retournée dans le signal ERR_L_RQFAILED de l'entité de signalisation locale à l'utilisateur AAL. Si la cause de la défaillance n'est pas 49 (Qualité de service indisponible), 51 (débit de cellules non disponible chez l'utilisateur - UNI 3.0), 37 (débit de cellules non disponible chez l'utilisateur - UNI 3.1), ou 41 (Défaillance temporaire) l'adresse ATM du point d'extrémité est éliminée de l'ensemble {ATM.1, ATM.2, ..., ATM.n} retourné par le MARS. Autrement, le L_MULTI_RQ ouL_MULTI_ADD devrait être réémis après un

délai aléatoire de 5 à 10 secondes. Si la demande échoue à nouveau, une autre demande devrait être produite après deux fois l'écoulement de deux fois le délai précédent. Ce processus devrait se continuer jusqu'à ce que l'appel réussisse ou que le VC multipoint soit libéré.

Si le L_MULTI_RQ initial échoue pour ATM.1, et si n est supérieur à 1 (c'est-à-dire, si l'ensemble d'adresses ATM retourné contient deux adresses ou plus) un nouveau L_MULTI_RQ devrait être immédiatement produit pour la prochaine adresse ATM dans l'ensemble. Cette procédure est répétée jusqu'à ce qu'un L_MULTI_RQ réussisse, car aucun L_MULTI_ADD ne peut être produit tant que n'est pas établi un VC sortant initial. Chaque adresse ATM pour laquelle un L_MULTI_RQ a échoué avec la cause 49, 51, 37, ou 41 DOIT être marqué plutôt que supprimé. Un L_MULTI_ADD est produit pour ces adresses étiquetées en utilisant la procédure de délai aléatoire mentionnée ci-dessus.

Le VC PEUT être considéré comme "ouvert" avant que L_MULTI_ADD aient été réémis avec succès. Un point d'extrémité PEUT mettre en œuvre un mécanisme concurrent qui permette aux données de commencer à s'écouler du nouveau VC même si L_MULTI_ADD échoués ont été réessayés. (C'est une solution de remplacement plutôt que d'attendre que chaque nœud d'extrémité ait accepté la connexion ce qui pourrait conduire à des délais significatifs pour la transmission du premier paquet.)

Chaque VC DOIT avoir associé un temporisateur d'inactivité configurable. Si le temporisateur arrive à expiration, un L_RELEASE est produit pour ce VC, et l'adresse de classe D n'est plus considérée comme ayant un chemin actif de sortie de l'hôte local. Le temporisateur DEVRAIT n'être pas inférieur à 1 minute, et une valeur par défaut de 20 minutes est RECOMMANDÉE. Le choix de périodes de temporisation spécifiques sort du domaine d'application du présent document.

La consommation de VC peut aussi être réduite par le fait que les points d'extrémité notent quand un nouvel ensemble de {ATM.1, ...ATM.n} du groupe correspond à celui d'un VC préexistant sortant d'un autre groupe. Avec une gestion locale soigneuse, et en supposant que la QS du VC existant est suffisante pour les deux groupes, un nouveau VC point à multipoint peut n'être pas nécessaire. Dans certaines circonstances, des points d'extrémité peuvent décider qu'il est suffisant de réutiliser un VC existant dont l'ensemble de nœuds d'extrémité est un surensemble des membres du nouveau groupe (auquel cas, certains points d'extrémité vont recevoir du trafic de diffusion groupée pour un groupe de couche 3 qu'ils ont rejoint, et doivent le filtrer au dessus de l'interface ATM). Les algorithmes pour effectuer ce type d'optimisation ne sont pas discutés ici, et ne sont pas exigés pour la conformité au présent document.

5.1.4 Traçage des mises à jour suivantes du groupe

Une fois qu'un nouveau VC a été établi, le côté émetteur de l'interface de membre de la grappe doit surveiller les changements ultérieurs du groupe – ajoutant ou éliminant en tant que de besoin les nœuds d'extrémité. Ceci est réalisé en surveillant les messages MARS_JOIN et MARS_LEAVE provenant du MARS lui-même. Ces messages sont décrits en détail au paragraphe 5.2 – pour l'instant, il est suffisant de noter qu'ils portent :

- l'adresse ATM d'un nœud qui se joint ou quitte un groupe,
- l'adresse de couche 3 du ou des groupes rejoints ou quittés,
- un numéro de séquence de grappe (CSN, *Cluster Sequence Number*) provenant du MARS.

Les messages MARS_JOIN et MARS_LEAVE arrivent à chaque membre de la grappe par le ClusterControlVC. Les messages MARS_JOIN ou MARS_LEAVE qui confirment simplement les informations déjà détenues par le membre de la grappe sont utilisés pour suivre la trace du numéro de séquence de grappe, mais sont ignorés par ailleurs.

5.1.4.1 Mise à jour des VC actifs

Si on voit un MARS_JOIN qui se réfère à (ou englobe) un groupe pour lequel le côté émission a déjà un VC ouvert, l'adresse ATM du nouveau membre est extraite et un L_MULTI_ADD est produit localement. Cela assure que les points d'extrémité qui envoient déjà à un groupe donné vont immédiatement ajouter le nouveau membre à la liste de leurs receveurs.

Si on voit un MARS_LEAVE qui se réfère à (ou englobe) un groupe pour lequel le côté émission a déjà un VC ouvert, l'adresse ATM de l'ancien membre est extraite et un L_MULTI_DROP est produit localement. Cela assure que les points d'extrémité qui envoient déjà à un groupe donné vont immédiatement éliminer le vieux membre de la liste de leurs receveurs. Lorsque la dernière extrémité d'un VC est abandonnée, le VC est fermé complètement et le groupe affecté n'a plus de chemin de sortie du point d'extrémité local (le prochain paquet sortant pour l'adresse de ce groupe va déclancher la création d'un nouveau VC, comme décrit des paragraphes 5.1.1 à 5.1.3).

Le côté émission de l'interface NE DOIT PAS fermer un VC actif pour un groupe pour lequel le côté réception a juste exécuté un LeaveLocalGroup. (Ce comportement est cohérent avec le modèle des hôtes qui transmettent à des groupes sans considération de leur propre statut de membre.)

Si un MARS_JOIN ou MARS_LEAVE arrive avec mar$pnum == 0, il ne porte pas de paire <min,max>, et n'est utilisé que pour suivre le CSN.

5.1.4.2 Suivi du numéro de séquence de grappe

Il est important que les points d'extrémité ne manquent pas les mises à jour des adhésions au groupe produites par le MARS sur ClusterControlVC. Cependant, cela va arriver de temps en temps. Le numéro de séquence de grappe est porté par une valeur non signée de 32 bits dans le champ mar$msn de nombreux messages du MARS (sauf par MARS_REQUEST et MARS_NAK). EIle s'incrémente d'un à chaque transmission de ClusterControlVC par le MARS, sans considérer si la transmission représente ou non un changement dans la base de données du MARS. En suivant ce compteur, les membres de la grappe peuvent déterminer si ils ont manqué un message précédent sur ClusterControlVC, et un éventuel changement des adhésions. Ceci est utilisé pour déclancher la revalidation (décrite au paragraphe 5.1.5).

Le CSN actuel est copié dans le champ mar$msn des messages du MARS envoyés aux membres de la grappe, qu'ils soient sur un ClusterControlVC ou sur un VC en point à point.

Les calculs sur les numéros de séquence DOIVENT être effectués sur une arithmétique de valeurs non signées de 32 bits.

Chaque membre de grappe conserve son propre numéro de séquence d'hôte (HSN, *Host Sequence Number*) de 32 bits pour garder trace du numéro de séquence du MARS. Chaque fois qu'est reçu un message portant un champ mar$msn, le traitement suivant est effectué :

```
Seq.diff = mar$msn - HSN

mar$msn -> HSN
{...traiter le message du MARS comme approprié...}

si ((Seq.diff != 1) && (Seq.diff != 0))
  alors {...revalider les informations d'adhésion au groupe...}
```

Le résultat de base est que le membre de la grappe essaye de rester en phase avec les changements d'adhésion notés par le MARS. Si jamais il détecte qu'est survenu un changement d'adhésion (dans tout groupe) sans qu'il l'ait remarqué, il re-valide l'adhésion à tous les groupes avec lesquels il a actuellement des VC de diffusion groupée ouverts.

La valeur mar$msn dans un MARS_MULTI individuel n'est pas utilisée pour mettre à jour le HSN jusqu'à ce que toutes les parties du MARS_MULTI (s'il y en a plus d'une) soient arrivées. (Si le mar$msn change, le MARS_MULTI est éliminé, comme décrit au paragraphe 5.1.1.)

Le MARS est libre de choisir une valeur initiale de CSN. Lorsque un nouveau membre de la grappe démarre, il devrait initialiser le HSN à zéro. Lorsque le membre de la grappe envoie le MARS_JOIN pour s'enregistrer (c'est décrit plus loin) le HSN sera correctement mis à jour à la valeur de CSN actuelle lorsque le point d'extrémité reçoit la copie de son MARS_JOIN renvoyée par le MARS.

5.1.5 Revalidation des nœuds d'extrémité d'un VC

Certains événements peuvent informer un membre de grappe qu'il a des informations incorrectes sur les ensembles de nœuds d'extrémité auxquels il devrait envoyer. Si une erreur survient sur un VC associé à un groupe particulier, le membre de la grappe initie une procédure de revalidation pour ce groupe spécifique. Si un trou est détecté dans les numéros de séquence de grappe, cela initie une revalidation de tous les groupes auxquels le membre de la grappe a actuellement des VC en point à multipoint ouverts.

Chaque VC multipoint ouvert et actif a un fanion associé appelé "VC_revalidate". Ce fanion est vérifié chaque fois qu'un paquet est mis en file d'attente pour transmission sur ce VC. Si le fanion est faux, le paquet est transmis et aucune autre action n'est requise.

Cependant, si le fanion VC_revalidate est vrai, le paquet est alors transmis et une nouvelle séquence d'événements est commencée localement.

La revalidation commence par la réémission d'une MARS_REQUEST pour le groupe à revalider. L'ensemble de membres retourné {NewATM.1, NewATM.2, ... NewATM.n} est comparé à l'ensemble déjà détenu localement. Des L_MULTI_DROP sont produits sur le VC du groupe pour chaque nœud qui apparaît dans l'ensemble d'origine des membres mais pas dans l'ensemble revalidé des membres. Des L_MULTI_ADD sont produits sur le VC du groupe pour chaque nœud qui apparaît dans l'ensemble revalidé des membres mais pas dans l'ensemble d'origine des membres. Le

fanion VC_revalidate est remis à zéro quand la revalidation se termine pour le groupe en question. Les mécanismes spécifiques de la mise en œuvre auront besoin de mettre le fanion dans l'état "revalidation en cours".

La différence clé entre la construction d'un VC (paragrahe 5.1.3) et la revalidating d'un VC est que la transmission de paquet continue sur le VC ouvert alors qu'il est en cours de revalidation. Cela minimise les perturbations du trafic existant.

L'algorithme pour initier la revalidation est :

- Lorsque un paquet arrive pour transmission sur un groupe donné, l'adhésion au groupe est revalidée si VC_revalidate == VRAI.
  La revalidation remet VC_revalidate à zéro.
- Lorsque survient un événement qui demande revalidation, chaque groupe a son fanion VC_revalidate mis à VRAI à un moment aléatoire entre 1 et 10 secondes.

Avantage : la revalidation des groupes actifs survient rapidement, et les groupes essentiellement oisifs sont revalidés en tant que de besoin. La répartition aléatoire du réglage des fanions VC_revalidate améliore les chances d'échelonner les demandes de revalidation provenant des envoyeurs lorsque est détecté un trou dans les numéros de séquence.

5.1.5.1 Quand un nœud d'extrémité s'élimine lui-même

Durant la vie d'un VC multipoint un ERR_L_DROP peut être reçu pour indiquer qu'un nœud d'extrémité a terminé sa participation au niveau ATM. Le point d'extrémité ATM associé au ERR_L_DROP DOIT être retiré de l'ensemble {ATM.1, ATM.2, ... ATM.n} détenu localement associé au VC.

Après un délai aléatoire entre 1 et 10 secondes, le VC_revalidate du fanion associé à ce VC DOIT être mis à VRAI.

Si un ERR_L_RELEASE est reçu, l'ensemble {ATM.1, ATM.2, ... ATM.n} entier est alors éliminé et le VC est considéré comme complètement fermé. Les transmissions de paquets ultérieures au groupe desservi par ce VC vont résulter en l'établissement d'un nouveau VC, comme décrit au paragraphe 5.1.3.

5.1.5.2 Quand un trou est détecté dans le CSN.

Le paragraphe 5.1.4.2 décrit comment un trou du CSN est détecté. Si un trou du CSN est détecté à réception d'un MARS_JOIN ou d'un MARS_LEAVE, tout VC de diffusion groupée sortant DOIT avoir son fanion VC_revalidate réglé à VRAI à un intervalle aléatoire entre 1 et 10 secondes à partir de la détection du trou du CSN.

La seule exception à cette règle est lorsque un trou est détecté dans les numéros de séquence durant l'établissement du VC d'un nouveau groupe (c'est-à-dire, une réponse MARS_MULTI a été reçue correctement, mais son mar$msn indiquait que du trafic de MARS antérieur a été manqué sur ClusterControlVC). Dans ce cas, tout VC ouvert, EXCEPTÉ celui qui vient d'être établi, DOIT avoir son fanion VC_revalidate à vrai à un intervalle aléatoire entre 1 et 10 secondes du moment où le trou du CSN a été détecté. (Le VC qui vient d'être établi à ce moment est considéré comme déjà validé.)

5.1.6 "Migration" du VC multipoint sortant

En plus du traçage de groupe décrit au paragraphe 5.1.4, le côté émission d'un membre de grappe doit répondre aux demandes de "migration" du MARS. Elles sont déclanchées par la réception d'un message MARS_MIGRATE provenant du ClusterControlVC. Le message MARS_MIGRATE est décrit ci-dessous, avec un code mar$op de 13.

Données :

mar$afn	16 bits	Famille d'adresse (0x000F).
mar$pro	56 bits	Identification du protocole.
mar$hdrrsv	24 bits	Réservé. Non utilisé par le protocole de contrôle de MARS.
mar$chksum	16 bits	Somme de contrôle sur le message MARS entier.
mar$extoff	16 bits	Décalage d'extensions.
mar$op	16 bits	Code de fonctionnement (MARS_MIGRATE = 13).
mar$shtl	8 bits	Type & longueur du numéro ATM de source. (r)
mar$sstl	8 bits	Type & longueur de la sous adresse ATM de source. (q)
mar$spln	8 bits	Longueur de l'adresse de source du protocole. (s)
mar$thtl	8 bits	Type & longueur du numéro ATM cible. (x)
mar$tstl	8 bits	Type & longueur de la sous adresse ATM cible. (y)
mar$tpln	8 bits	Longueur de l'adresse du groupe cible. (z)
mar$tnum	16 bits	Nombre d'adresses ATM cibles retournées. (N)
mar$resv	16 bits	Réservé.
mar$msn	32 bits	Numéro de séquence MARS.

mar$sha	roctets	Numéro ATM de source.
mar$ssa	qoctets	Sous adresse ATM de source.
mar$spa	soctets	Adresse de source du protocole.
mar$tpa	zoctets	Adresse cible du groupe de diffusion groupée.
mar$tha.1	xoctets	Cible ATM numéro 1
mar$tsa.1	yoctets	Sous adresse cible ATM 1
mar$tha.2	xoctets	Cible ATM numéro 2
mar$tsa.2	yoctets	Sous adresse cible ATM 2
[.....]		
mar$tha.N	xoctets	Cible ATM numéro N
mar$tsa.N	yoctets	Sous adresse cible N

Une migration est demandée lorsque le MARS détermine qu'il ne veut plus que les membres de la grappe transmettent directement leurs paquets aux adresses ATM qu'il avait précédemment spécifié (par les MARS_REQUEST ou les MARS_JOIN). Lorsqu'il reçoit une MARS_MIGRATE, chaque membre de la grappe DOIT effectuer les étapes suivantes :

Fermer tout VC sortant existant associé au groupe porté dans le champ mar$tpa (L_RELEASE), ou dissocier le groupe de tout VC sortant qu'il aurait pu partager (comme décrit au paragraphe 5.1.3).

Établir un nouveau VC sortant pour le groupe spécifié, en utilisant l'algorithme décrit au paragraphe 5.1.3 et en prenant l'ensemble d'adresses ATM fourni dans le MARS_MIGRATE comme nouvel ensemble de membres du groupe {ATM.1, ... ATM.n}.

Le MARS_MIGRATE porte le nouvel ensemble de membres {ATM.1, ... ATM.n} dans un seul message, de façon similaire à celle d'un MARS_MULTI en une seule partie. Comme avec les autres messages provenant du MARS, le numéro de séquence de grappe porté dans mar$msn est vérifié comme décrit au paragraphe 5.1.4.2.

5.2 Comportement du côté réception

Un membre d'une grappe est un "membre du groupe" (au sens où il reçoit des paquets dirigés sur un groupe de diffusion groupée donné) lorsque son adresse ATM apparaît dans l'entrée de tableau du MARS pour l'adresse de diffusion groupée du groupe. Une fonction clé au sein de chaque grappe est la distribution aux membres de la grappe des informations provenant du MARS sur les membres du groupe.

Un point d'extrémité peut souhaiter se "joindre à un groupe" en réponse à une demande locale, de niveau supérieur, d'adhésion à un groupe ou parce que le point d'extrémité prend en charge un moteur de transmission de diffusion groupée de couche 3 qui exige la capacité de "voir" le trafic intra-grappe afin de le transmettre.

Deux messages prennent en charge ces exigences - MARS_JOIN et MARS_LEAVE. Ils sont envoyés au MARS par les points d'extrémité lorsque il est demandé à la couche 3 locale/interface ATM de se joindre ou de quitter un groupe de diffusion groupée. Le MARS propage ces messages en retour sur le ClusterControlVC, pour s'assurer que la connaissance du changement de composition du groupe est distribuée à temps aux autres membres de la grappe.

Certains modèles de points d'extrémité de couche 3 (par exemple, des routeurs de diffusion groupée IP) s'attendent à être capables de recevoir du trafic de paquet "disparate" à travers tous les groupes. Cette fonctionnalité peut être émulée en permettant aux routeurs de demander que le MARS leur renvoie en retour à titre de membres "génériques" de toutes les adresses de classe D. Cependant, un problème inhérent au modèle ATM actuel est qu'un routeur complètement disparate peut épuiser les ressources locales de réassemblage dans son interface ATM. MARS_JOIN prend en charge une généralisation de la notion d'entrées "génériques", permettant aux routeurs de se limiter aux "blocs" de l'espace d'adresse de classe D. L'utilisation de cette facilité est décrite plus en détail à la Section 8.

Un bloc peut être de 1 (un seul groupe) ou de la taille de l'espace d'adresses de diffusion groupée tout entier (par exemple, le comportement "disparate" IPv4 'par défaut). Un bloc est défini comme toutes les adresses entre, et y compris, une paire d'adresses <min,max>. Un MARS_JOIN ou MARS_LEAVE peut porter plusieurs paires <min,max>.

Les membres d'une grappe DOIVENT fournir SEULEMENT une paire <min,max> pour chaque message JOIN/LEAVE qu'ils produisent. Cependant, ils DOIVENT être capables de traiter plusieurs paires <min,max> dans les messages JOIN/LEAVE quand ils effectuent la gestion de VC comme décrit au paragraphe 5.1.4 (l'interprétation étant que l'opération adhésion/départ s'applique à toutes les adresses dans la gamme de <min> à <max> inclus, pour chaque paire <min,max>).

Dans les environnements de la RFC1112, un MARS_JOIN pour un seul groupe est déclanché par un signal JoinLocalGroup provenant de la couche IP. Un MARS_LEAVE pour un seul groupe est déclanché par un signal LeaveLocalGroup provenant de la couche IP.

Les membres d'une grappe qui ont des exigences particulières (par exemple, les routeurs de diffusion groupée) peuvent produire des MARS_JOIN et des MARS_LEAVE qui spécifient un seul bloc de deux adresses de groupes de diffusion groupée ou plus. Cependant, un membre de grappe NE DEVRA PAS produire une telle adhésion de bloc multi groupes pour une gamme d'adresses entièrement ou partiellement recouverte par une ou des adhésions de bloc multi groupes que le membre de la grappe avait précédemment produites et non encore rétractées. Un membre de grappe PEUT produire des combinaisons de MARS_JOIN de groupes seuls qui se recouvrent avec un MARS_JOIN de bloc multi groupes.

Un point d'extrémité DOIT s'enregistrer auprès d'un MARS afin de devenir membre d'une grappe et être ajouté comme extrémité au ClusterControlVC. L'enregistrement est traité au paragraphe 5.2.3.

Finalement, le point d'extrémité DOIT être capable de terminer les VC unidirectionnels (c'est-à-dire, agir comme un nœud d'extrémité d'un VC point à multipoint UNI 3.0/3.1, avec une bande passante de zéro allouée au chemin de retour). La RFC1755 décrit les informations de signalisation requises pour terminer les VC qui portent du trafic LLC/SNAP encapsulé (exposé au paragraphe 5.5).

5.2.1 Format des messages MARS_JOIN et MARS_LEAVE

Le message MARS_JOIN est indiqué par une valeur de type d'opération de 4. MARS_LEAVE a le même format et la valeur de type d'opération de 5. Le format du message est :

Données :		
mar$afn	16 bits	Famille d'adresses (0x000F).
mar$pro	56 bits	Identification du protocole.
mar$hdrrsv	24 bits	Réservé. Non vu du protocole de contrôle de MARS.
mar$chksum	16 bits	Somme de contrôle sur le message MARS entier.
mar$extoff	16 bits	Décalage des extensions.
mar$op	16 bits	Code d'opération (MARS_JOIN ou MARS_LEAVE).
mar$shtl	8 bits	Type & longueur du numéro ATM de source. (r)
mar$sstl	8 bits	Type & longueur de la sous adresse ATM de source. (q)
mar$spln	8 bits	Longueur de l'adresse du protocole de source. (s)
mar$tpln	8 bits	Longueur de l'adresse de groupe. (z)
mar$pnum	16 bits	Nombre de paires d'adresses de groupe. (N)
mar$flags	16 bits	Fanions de groupe de couche 3, copie, et enregistrer.
mar$cmi	16 bits	Identifiant de membre de grappe.
mar$msn	32 bits	Numéro de séquence MARS.
mar$sha	roctets	Numéro ATM de source.
mar$ssa	qoctets	Sous adresse ATM de source.
mar$spa	soctets	Adresse du protocole de source.
mar$min.1	zoctets	Minimum d'adresse de groupe de diffusion groupée - paire.1
mar$max.1	zoctets	Maximum d'adresse de groupe de diffusion groupée - paire.1
[.....]		
mar$min.N	zoctets	Minimum d'adresse de groupe de diffusion groupée - paire.N
mar$max.N	zoctets	Maximum d'adresse de groupe de diffusion groupée - paire.N

mar$spln indique le nombre d'octets dans l'adresse de protocole du point d'extrémité de source, et est interprété dans le contexte du protocole indiqué par le champ mar$pro. (Par exemple, dans les environnements IPv4, mar$pro sera 0x800, mar$spln est 4, et mar$tpln est 4.)

Le champ mar$flags contient trois fanions :

Bit 15 - mar$flags.layer3grp.
Bit 14 - mar$flags.copy.
Bit 13 - mar$flags.register.
Bit 12 - mar$flags.punched.
Bit 0-7 - mar$flags.sequence.
Bits 8 à 11 – réservé, DOIVENT être à zéro.

mar$flags.sequence est établi par les membres de la grappe, et DOIT toujours être passé non modifié par le MARS lors de la retransmission des messages MARS_JOIN ou MARS_LEAVE. Il est spécifique de la source, et DOIT être ignoré des autres membres de la grappe. Son utilisation est décrite au paragraphe 5.2.2.

mar$flags.punched DOIT être zéro lorsque le message MARS_JOIN ou MARS_LEAVE est transmis au MARS. Son utilisation est décrite aux paragraphes 5.2.2 et 6.2.4.

mar$flags.copy DOIT être réglé à 0 lorsque le message est envoyé d'un client MARS, et DOIT être mis à 1 lorsque le message est envoyé d'un MARS. (Ce fanion est destiné à prendre en charge la fonction MARS avec un des clients MARS dans la grappe. La destination d'un MARS_JOIN entrant peut être déterminée à partir de sa valeur.)

mar$flags.layer3grp permet au MARS de fournir les informations sur l'appartenance au groupe décrites au paragraphe 5.3. Les règles de son utilisation sont :

mar$flags.layer3grp DOIT ëtre réglé lorsque le membre de la grappe produit le MARS_JOIN comme résultat d'une adhésion explicite à un groupe de diffusion groupée de couche 3. (Par exemple, résultant d'une opération JoinHostGroup chez un hôte conforme à la RFC1112).

mar$flags.layer3grp DOIT être reréglé dans chaque MARS_JOIN si le MARS_JOIN est simplement l'enregistrement d'interface ip/atm locale à recevoir du trafic sur ce groupe pour des raisons qui lui sont propres.

mar$flags.layer3grp est ignoré et DOIT être traité comme reréglé par le MARS pour tout MARS_JOIN qui spécifie un bloc couvrant plus d'un seul groupe (par exemple, un jonction à un bloc provenant d'un routeur pour s'assurer que les moteurs de transmission "voient" tout le trafic).

mar$flags.register indique si le MARS_JOIN ou MARS_LEAVE va être utilisé pour enregistrer ou désenregistrer un membre de la grappe (décrit au paragraphe 5.2.3). Lorsque utilisé pour se joindre ou quitter un groupe spécifique, le fanion mar$register DOIT être zéro.

mar$pnum indique combien de paires <min,max> sont incluses dans le message. Ce champ DOIT être à 1 quand le message est envoyé d'un membre de a grappe. Un MARS PEUT retourner un MARS_JOIN ou MARS_LEAVE avec toute valeur de mar$pnum, y compris zéro. Cela sera expliqué au paragraphe 6.2.4.

Le champ mar$cmi DOIT être mis à zéro par les membres de la grappe, et il est utilisé par le MARS durant l'enregistrment des membres de la grappe, selon la description du paragraphe 5.2.3.

mar$msn DOIT être mis à zéro lorsqu'il est transmis par un point d'extrémité. Il est réglé à la valeur actuelle du numéro de séquence de grappe par le MARS lors de la retransmission due MARS_JOIN ou MARS_LEAVE. Son utilisation a été décrite au paragraphe 5.1.4.

Pour simplifier la construction et l'analyse des messages MARS_JOIN et MARS_LEAVE, les restrictions suivantes sont imposées aux paires <min,max> :

Supposer que max(N) est le champ <max> de la $N^{ème}$ paire <min,max>.
Supposer que min(N) est le champ <min> de la $N^{ème}$ paire <min,max>.
Supposer qu'un message se joindre/quitter arrive avec K paires <min,max>.
Ce qui suit doit être vrai :
$max(N) < min(N+1)$ pour $1 \leq\leq N < K$
$max(N) \geq min(N)$ pour $1 \leq N \leq K$

En langage clair, l'ensemble doit spécifier une séquence ascendante de blocs d'adresses. La définition de "supérieur à" ou "inférieur à" peut être spécifique du protocole. Dans les environnements IPv4, les adresses sont traitées comme des valeurs binaires non signées de 32 bits (octet de poids fort en premier).

5.2.1.1 Importantes valeurs IPv4 par défaut

Les opérations JoinLocalGroup et LeaveLocalGroup ne sont valides que pour un seul groupe. Pour toute adresse arbitraire de groupe X, le MARS_JOIN ou MARS_LEAVE associé DOIT spécifier une seule paire <X, X>. mar$flags.layer3grp DOIT être établi dans ces circonstances.

Un routeur qui choisit de se comporter en stricte conformété avec la RFC1112 DOIT spécifier l'espace de classe D entier. Le MARS_JOIN ou MARS_LEAVE associé DOIT spécifier une seule paire <224.0.0.0, 239.255.255.255>. Chaque fois

qu'un routeur produit un MARS_JOIN dans le seul but de transmettre du trafic IP, il DOIT remettre mar$flags.layer3grp à zéro.

L'utilisation de valeurs de <min, max> de remplacement par les routeurs de diffusion groupée est exposée à la Section 8.

5.2.2 Retransmission des messages MARS_JOIN et MARS_LEAVE

Des problèmes transitoires peuvent résulter en la perte de messages entre le MARS et les membres de la grappe.

Un algorithme simple est utilisé pour résoudre ce problème. Les membres de la grappe retransmettent chaque message MARS_JOIN et MARS_LEAVE à intervalle régulier jusqu'à ce qu'ils reçoivent à nouveau une copie en retour, soit sur ClusterControlVC, soit sur le VC sur lequel ils envoient le message. À ce moment, le point d'extrémité local peut être certain que le MARS l'a reçu et traité.

L'intervalle ne devrait pas être inférieur à 5 secondes, et une valeur par défaut de 10 secondes est recommandée. Après cinq retransmissions, la tentative devrait être marquée localement comme un échec. Ceci DOIT être considéré comme une défaillance du MARS, et déclancher la reconnexion de MARS décrite au paragraphe 5.4.

Une "copie" est définie comme un message reçu avec les champs suivant qui correspondent à ceux d'un MARS_JOIN/LEAVE transmis précédemment :

- mar$op
- mar$flags.register
- mar$flags.sequence
- mar$pnum
- adresse ATM de source
- première paire <min,max>

De plus, une copie valide DOIT avoir les valeurs de champ suivantes :

- mar$flags.punched = 0
- mar$flags.copy = 1

Le champ mar$flags.sequence n'est jamais modifié ou vérifié par un MARS. Les mises en œuvre PEUVENT choisir d'utiliser localement des schémas de numéro de séquence significatifs, qui PEUVENT différer d'un membre de la grappe à l'autre. En l'absence de tels schémas, la valeur par défaut pour mar$flags.sequence DOIT être zéro.

Des mises en œuvre prudentes PEUVENT avoir plus d'un MARS_JOIN/LEAVE non acquitté en cours à la fois.

5.2.3 Enregistrement et désenregistrement des memebres de la grappe

Pour devenir membre de la grappe, un point d'extrémité doit s'enregistrer auprès du MARS. Cela réalise deux choses – le point d'extrémité est ajouté comme nœud d'extrémité au ClusterControlVC, et le point d'extrémité reçoit un identifiant de membre de grappe (CMI) de 16 bits. Le CMI identifie de façon univoque chaque point d'extrémité qui est rattaché à la grappe.

L'enregistrement auprès du MARS survient lorsque un point d'extrémité produit un MARS_JOIN avec le fanion mar$flags.register réglé à un (bit 13 du champ mar$flags).

Le membre de grappe DOIT inclure son adresse ATM de source, et PEUT choisir de spécifier lors de l'enregistrement une adresse de protocole de source nulle.

Aucune adresse de groupe spécifique d'un protocole n'est incluse dans un enregistrement MARS_JOIN.

Le membre de la grappe retransmet ce MARS_JOIN conformément au paragraphe 5.2.2 jusqu'à ce qu'il confirme que le MARS l'a reçu.

Lorsque l'enregistrement MARS_JOIN est retourné, il contient une valeur différente de zéro dans mar$cmi. Cette valeur DOIT être notée par le membre de la grappe, et utilisée chaque fois que les circonstances exigent le CMI du membre de la grappe.

Un point d'extrémité peut aussi choisir de se désenregistrer, en utilisant un MARS_LEAVE avec mar$flags.register établi. Il en résultera que le MARS abandonne le point d'extrémité dans ClusterControlVC, retirant toute référence au membre dans la base de données des transpositions, et libérant son CMI.

Comme pour l'enregistrement, une demande de désenregistrement DOIT comporter l'adresse ATM de source correcte pour le membre de la grappe, mais PEUT choisir de spécifier une adresse de protocole de source nulle.

Le membre de la grappe retransmet ce MARS_LEAVE conformément au paragraphe 5.2.2 jusqu'à ce qu'il confirme que le MARS l'a reçu.

5.3 Prise en charge de la gestion de groupe de couche 3

Bien que l'intention de la présente spécification soit d'être indépendante des questions de couche 3, on essaye d'aider le fonctionnement des protocoles d'acheminement de diffusion groupée de couche 3 qui ont besoin de vérifier si des groupes ont des membres au sein d'une grappe.

Dans l'exemple de IP, où IGMP est simplement utilisé (comme décrit à la section 2) pour déterminer si d'autres membres de la grappe écoutent un groupe parce qu'ils ont des applications de couche supérieure qui veulent recevoir le trafic d'un groupe.

Les routeurs peuvent choisir d'interroger le MARS sur ces informations, plutôt que de faire des interrogations IGMP en diffusion groupée au 224.0.0.1 et de supporter les coûts associés pour l'établissement d'un VC avec tous les systèmes de la grappe.

L'interrogation est produite par l'envoi d'une MARS_GROUPLIST_REQUEST au MARS. La MARS_GROUPLIST_REQUEST est construite à partir d'un MARS_JOIN, mais elle a un code de fonctionnement de 10. la première paire <min,max> sera utilisée par le MARS pour identifier la gamme des groupes auxquels le membre de grappe qui interroge s'intéresse. Toutes paires <min,max> supplémentaires seront ignorées. Une demande avec mar$pnum = 0 sera ignorée.

La réponse du MARS est uneMARS_GROUPLIST_REPLY, qui porte une liste des groupes de diffusion groupée qui au sein du bloc <min,max> spécifié ont des membres de couche 3. Un groupe est noté dans cette liste si un ou plusieurs des MARS_JOIN qui ont généré son entrée de transposition dans le MARS contenait un fanion mar$flags.layer3grp établi.

Les réponses MARS_GROUPLIST_REPLY sont transmises en retour au membre de grappe auteur de l'interrogation sur le VC utilisé pour envoyer la demande MARS_GROUPLIST_REQUEST.

La réponse MARS_GROUPLIST_REPLY est déduite de MARS_MULTI mais avec mar$op = 11. Elle peut avoir plusieurs parties si nécessaire, et elle est reçue de façon similaire à celle du MARS_MULTI.

Données :

mar$afn	16 bits	Famille d'adresses (0x000F).
mar$pro	56 bits	Identification du protocole.
mar$hdrrsv	24 bits	Réservé. Non utilisé par le protocole de contrôle de MARS.
mar$chksum	16 bits	Somme de contrôle sur le message MARS entier.
mar$extoff	16 bits	Décalage des extensions.
mar$op	16 bits	Code de fonctionnement (MARS_GROUPLIST_REPLY).
mar$shtl	8 bits	Type & longueur du numéro ATM de source. (r)
mar$sstl	8 bits	Type & longueur de la sous-adresse ATM de source. (q)
mar$spln	8 bits	Longueur de l'adresse du protocole de source. (s)
mar$thtl	8 bits	Non utilisé – mis à zéro.
mar$tstl	8 bits	Non utilisé – mis à zéro.
mar$tpln	8 bits	Longueur de l'adresse du groupe cible. (z)
mar$tnum	16 bits	Nombre des adresses de groupe retournées. (N).
mar$seqxy	16 bits	Fanion booléen x et numéro de séquence y.
mar$msn	32 bits	Numéro de séquence MARS.
mar$sha	roctets	Numéro de source ATM.
mar$ssa	qoctets	Sous adresse ATM de source.
mar$spa	soctets	Adresse de protocole de source.
mar$mgrp.1	zoctets	Adresse de groupe 1
[.....]		
mar$mgrp.N	zoctets	Adresse de groupe N

mar$seqxy est codé comme pour le multiple MARS_MULTI -

Les composants de MARS_GROUPLIST_REPLY sont transmis et reçus en utilisant le même algorithme que décrit au paragraphe 5.1.1 pour MARS_MULTI. La seule différence est que les adresses de protocole sont retournées à la place des adresses ATM.

Comme avec les MARS_MULTI, si une erreur survient dans la réception d'une réponse MARS_GROUPLIST_REPLY multi parties, l'ensemble de la réponse DOIT être éliminé et la demande MARS_GROUPLIST_REQUEST émise à nouveau. (Cela inclut que la valeur mar$msn reste constante.)

Noter que la capacité à générer des messages MARS_GROUPLIST_REQUEST et à recevoir des messages MARS_GROUPLIST_REPLY n'est pas exigée pour les mises en œuvre d'interface d'hôte générales. Il est facultatif que les mises en œuvre d'interfaces prennent en charge les moteurs de transmission de diffusion groupée de couche 3. Cependant, cette fonctionnalité DOIT être prise en charge par le MARS.

5.4 Prise en charge des entités MARS redondantes ou de sauvegarde

Les points d'extrémité sont supposés avoir été configurés avec l'adresse ATM d'au moins un MARS. Les points d'extrémité PEUVENT choisir de tenir un tableau des adresses ATM, représentant des MARS de remplacement qui seront contactés au cas où le fonctionnement normal avec le MARS d'origine se révèlerait défaillant. On suppose que ce tableau range les adresses ATM en ordre décroissant de préférence.

Un point d'extrémité va normalement décider qu'il y a des problèmes avec le MARS lorsque :
- Il échoue à établir un VC en point à point avec le MARS.
- Les MARS_REQUEST échouent (paragraphe 5.1.1).
- Les MARS_JOIN/MARS_LEAVE échouent (paragraphe 5.2.2).
- Il n'a pas reçu de MARS_REDIRECT_MAP dans les quatre dernières minutes (paragraphe 5.4.3).

(Si il est capable de discerner quelle connexion représente ClusterControlVC, il peut aussi utiliser les défaillances de connexion sur ce VC pour indiquer des problèmes avec le MARS).

5.4.1 Première réponse aux problèmes de MARS

La première réponse est de supposer un problème transitoire avec le MARS qui est utilisé à ce moment. Le membre de grappe devrait attendre pendant une durée aléatoire entre 1 et 10 secondes avant de tenter de se reconnecter et de se réenregistrer auprès du MARS. Si l'enregistrement MARS_JOIN réussit, alors :

Le membre de grappe DOIT continuer à se joindre à tout groupe que son ou ses protocoles de couche supérieure locale ont rejoint. Il est recommandé qu'un délai aléatoire de 1 à 10 secondes soit inséré avant de tenter chaque MARS_JOIN.

Le membre de grappe DOIT initier la revalidation de chaque groupe de diffusion groupée auquel il envoyait (comme si un trou avait été détecté dans les numéros de séquence (paragraphe 5.1.5).

La procédure de jonction et de revalidation ne doit pas interrompre l'utilisation par le membre de grappe des VC en multipoint qui étaient déjà ouverts au moment de la défaillance du MARS.

Si le réenregistrement auprès du MARS actuel échoue, et si il n'y a pas d'adresse de MARS de sauvegarde configurée, le membre de la grappe DOIT attendre pendant au moins une minute avant de répéter la procédure de réenregistrement. Il est RECOMMANDÉ que le membre de la grappe signale une condition d'erreur d'une façon localement significative.

Cette procédure peut se répéter jusqu'à ce que les administrateurs du réseau interviennent manuellement ou que le MARS actuel revienne en fonctionnement normal.

5.4.2 Connexion à un MARS de sauvegarde

Si le réenregistrement auprès du MARS actuel échoue, et si d'autres adresses de MARS ont été configurées, on choisit l'adresse du MARS suivant sur la liste pour être le MARS actuel, et le membre de grappe redémarre immédiatement la procédure de réenregistrement décrite au paragraphe 5.4.1. Si elle réussit, le membre de la grappe va reprendre un fonctionnement normal en utilisant le nouveau MARS. Il est RECOMMANDÉ que le membre de la grappe signale la survenance de cette condition d'une façon localement significative.

Si la tentative de réenregistrement avec le nouveau MARS échoue, le membre de la grappe DOIT attendre au moins une minute avant de choisir la nouvelle adresse de MARS dans le tableau et de répéter la procédure. Si la fin du tableau a été atteinte, le membre de la grappe recommence au début du tableau (ce qui devrait être le MARS original avec lequel le membre de grappe avait commencé).

Le plus mauvais scénario va être celui où les membres de la grappe tournent en boucle à travers le tableau des adresses de MARS possibles jusqu'à intervention manuelle des administrateurs du réseau.

5.4.3 Listes de sauvegarde dynamiques, et redirections douces

Pour prendre en charge un certain niveau d'autoconfiguration, un message MARS a été défini pour permettre au MARS en cours de diffuser sur ClusterControlVC un tableau des adresses des MARS de sauvegarde. Lorsqu'ils reçoivent ce message, les membres de la grappe qui tiennent une liste d'adresses de MARS de sauvegarde DOIVENT insérer ces informations au début de leur liste tenue en local (c'est-à-dire que les informations fournies par le MARS ont une préférence supérieure à celle des adresses qui ont pu être configurées manuellement chez le membre de la grappe).

Le message est MARS_REDIRECT_MAP. Il se fonde sur le message MARS_MULTI, avec les modifications suivantes :

- le champ mar$tpln est remplacé par mar$redirf.
- le champ mar$spln est réservé.
- mar$tpa et mar$spa sont éliminés.

MARS_REDIRECT_MAP a un code de type de fonctionnement de 12 en décimal.

Données :

mar$afn	16 bits	Famille d'adresses (0x000F).
mar$pro	56 bits	Identification du protocole.
mar$hdrrsv	24 bits	Réservé. Non utilisé par le protocole de contrôle de MARS.
mar$chksum	16 bits	Somme de contrôle sur le message de MARS entier.
mar$extoff	16 bits	Décalage des extensions.
mar$op	16 bits	Code de fonctionnement (MARS_REDIRECT_MAP).
mar$shtl	8 bits	Type et longueur du numéro ATM de source. (r)
mar$sstl	8 bits	Type et longueur de la sous adresse ATM de source. (q)
mar$spln	8 bits	Longueur de l'adresse du protocole de source. (s)
mar$thtl	8 bits	Type et longueur du numéro ATM cible. (x)
mar$tstl	8 bits	Type et longueur de la sous-adresse ATM cible. (y)
mar$redirf	8 bits	Fanion de contrôle du comportement de redirection du client.
mar$tnum	16 bits	Nombre d'adresses de MARS retournées. (N).
mar$seqxy	16 bits	Fanion booléen x et numéro de séquence y.
mar$msn	32 bits	Numéro de séquence MARS.
mar$sha	roctets	Numéro ATM de source.
mar$ssa	qoctets	Sous -adresse ATM de source.
mar$tha.1	xoctets	Numéro ATM pour le MARS 1
mar$tsa.1	yoctets	Sous-adresse ATM pour le MARS 1
mar$tha.2	xoctets	Numéro ATM pour le MARS 2
mar$tsa.2	yoctets	Sous-adresse ATM pour le MARS 2
[.....]		
mar$tha.N	xoctets	Numéro ATM pour le MARS N
mar$tsa.N	yoctets	Sous-adresse ATM pour le MARS N

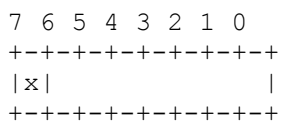
Le ou les champs d'adresse ATM de source DOIVENT identifier le MARS générateur. Un MARS_REDIRECT_MAP multi parties peut être transmis et réassemblé en utilisant le champ mar$seqxy de la même manière qu'un MARS_MULTI multi parties (paragraphe 5.1.1). Si une défaillance survient durant le réassemblage d'un MARS_REDIRECT_MAP multi parties (perte d'une partie, fin de temporisation de réassemblage, ou saut illégal de numéro de séquence MARS) le message entier DOIT être éliminé.

Ce message est transmis régulièrement par le MARS (il DOIT être transmis au moins toutes les deux minutes, il est RECOMMANDÉ qu'il soit transmis toutes les minutes).

Le MARS_REDIRECT_MAP est aussi utilisé pour forcer les membres de la grappe à passer d'un MARS à un autre. Si l'adresse ATM du premier MARS contenue dans un tableau MARS_REDIRECT_MAP n'est pas l'adresse du MARS en

cours d'un membre de la grappe, le client DOIT se "rediriger" sur le nouveau MARS. Le champ mar$redirf contrôle la façon dont se fait la redirection.

mar$redirf a le format suivant :



Si le bit 7 (le bit de poids fort) de mar$redirf est 1, le membre de grappe DOIT alors effectuer une redirection "dure". Ayant installé le nouveau tableau des adresses de MARS porté par MARS_REDIRECT_MAP, le membre de la grappe se réenregistre auprès du MARS qui est maintenant au sommet du tableau en utilisant le mécanisme décrit aux paragraphes 5.4.1 et 5.4.2.

Si le bit 7 de mar$redirf est 0, le membre de la grappe DOIT alors effectuer une redirection "douce", en commençant par les actions suivantes :
- ouvrir un VC point à point avec la première adresse ATM,
- tenter de s'enregistrer (paragraphe 5.2.3).

Si l'enregistrement réussit, le membre de la grappe clôt son VC point à point avec le MARS actuel (si il y en a un d'ouvert) puis procède à l'utilisation du nouveau VC point à point ouvert comme connexion au "MARS actuel". Le membre de la grappe N'ESSAYE PAS de rejoindre les groupes dont il est membre, ou de revalider les groupes auxquels il envoie.

C'est ce qu'on appelle une "redirection douce" parce qu'elle évite les traitements supplémentaires de jonction et de revalidation qui surviennent lorsque il y a récupération d'une défaillance de MARS. Cela suppose qu'il existe un mécanisme externe de synchronisation entre l'ancien et le nouveau MARS – méchanismes qui sortent du domaine d'application de la présente spécification.

Un certain niveau de confiance est requis avant l'initiation d'une redirection douce. Un membre de grappe DOIT vérifier que l'appelant à l'autre extrémité du VC sur lequel le MARS_REDIRECT_MAP est arrivé (en principe ClusterControlVC) est bien en fait le nœud qu'il attend comme MARS actuel.

D'autres applications de cette fonction feront l'objet d'études à l'avenir.

5.5 Encapsulations LLC/SNAP de chemin des données

Un schéma d'encapsulation étendu est requis pour la prise en charge du filtrage des éventuels paquets réfléchis (paragraphe 3.3).

Deux codets LLC/SNAP sont alloués à partir de l'espace OUI de l'IANA. Ils prennent en charge deux mécanismes différents pour la détection des paquets réfléchis. Ils sont appelés Encasulations de diffusion groupée de type n° 1 et de type n° 2.

Type n° 1
[0xAA-AA-03][0x00-00-5E][0x00-01][Paquet de couche 3 de type n° 1 étendu]
LLC OUI PID

Type n° 2
[0xAA-AA-03][0x00-00-5E][0x00-04][Paquet de couche 3 de type n° 2 étendu]
LLC OUI PID

Pour la conformité au présent document, les clients MARS :
DOIVENT transmettre les données en utilisant l'encapsulation de type n° 1.
DOIVENT être capables de recevoir correctement le trafic en utilisant l'encapsulation de type n° 1 OU de type n° 2.
NE DOIVENT PAS transmettre en utilisant l'encapsulation de type n° 2.

5.5.1 Encapsulation de type n° 1

Le paquet de couche 3 étendue de type n° 1 porte en son sein une copie de l'identifiant du membre de grappe (CMI) de la source et la "forme courte"' ou la "forme longue" du type de protocole selon le cas (paragraphe 4.3).

Lorsque il porte des paquets qui appartiennent à des protocoles qui ont des représentations de forme courte valides, le [paquet de couche 3 étendue de type n° 1] est codé par :

[pkt$cmi][pkt$pro][paquet de couche 3 original]
2 octets 2 octets N octets

Les deux premiers octets (pkt$cmi) portent le CMI alloué lorsque un point d'extrémité s'enregistre auprès du MARS (paragraphe 5.2.3). Les deux octets suivants (pkt$pro) indiquent le type de protocole du paquet porté dans le reste de la charge utile. Ceci est copié du champ mar$pro utilisé dans les messages MARS de contrôle.

Lorsque elle porte des paquets qui appartiennent à des protocoles qui n'ont qu'une forme longue de représentation (pkt$pro = 0x80) l'enveloppe DEVRA être encore étendue pour porter les 5 octets du champ mar$pro.snap (avec le bourrage pour un verrouillage sur 32 bits). La forme codée DEVRA être :

[pkt$cmi][0x00-80][mar$pro.snap][padding][Paquet original de couche 3]
2 octets 2 octets 5 octets 3 octets N octets

Le CMI est copié dans le champ pkt$cmi de chaque paquet de type n° 1 sortant. Lorsque une interface de point d'extrémité reçoit un AAL_SDU avec le codet LLC/SNAP qui indique l'encapsulation de type n° 1, elle compare le champ CMI avec son propre identifiant de membre de grappe pour le protocole indiqué. Le paquet est éliminé en silence si ils correspondent. Autrement, le paquet est accepté pour être traité par l'entité de protocole locale, identifiée par le ou les champs pkt$pro (et éventuellement SNAP).

Lorsque un protocole a des formes d'identification courte et longue valides, les receveurs PEUVENT choisir de reconnaître aussi la forme longue.

5.5.2 Encapsulation de type n° 2.

Des développements futurs pourraient permettre la diffusion groupée directe des AAL_SDU au-delà des frontières de grappes. L'expansion de l'ensemble des sources possibles dans ce sens pourrait être cause que le CMI devienne un paramètre inadéquat pour détecter les paquets réfléchis. Un champ plus large d'identification de source pourrait être nécessaire.

Le paquet de couche 3 étendue de type n° 2 porte en son sein un champ d'identifiant de source de 8 octets et soit la "forme courte", soit la "forme longue" du type de protocole selon le cas approprié (paragraphe 4.3). La forme et le contenu du champ ID de source ne sont pas actuellement spécifiés, et ne sont pertinents pour la conformité d'aucun client MARS selon le présent document. Les paquets encapsulés de type n° 2 reçus DOIVENT toujours être acceptés et passés à la couche supérieure indiquée par l'identifiant de protocole.

Lorsque il porte des paquets qui appartiennent à des protocoles qui ont des représentations de forme valide le [paquet de couche 3 étendue de type n° 2] est codé par :

[8 octet sourceID][mar$pro.type][Bourrage nul][Paquet de couche 3 original]
2 octets 2 octets

Lorsque il porte des paquets qui appartiennent à des protocoles qui ont seulement une représentation en forme longue (pkt$pro = 0x80) l'enveloppe DEVRA être encore étendue pour porter le champ mar$pro.snap de cinq octets (avec le bourrage pour le verrouillage sur 32 bits). La forme codée DEVRA être :

[8 octet sourceID][mar$pro.type][mar$pro.snap][Bourrage nul][Paquet de coucke 3]
2 octets 5 octets 1 octet

(Noter que dans ce cas, le bourrage après le champ SNAP est d'un octet au lieu des trois octets utilisés dans le type n° 1.) Lorsque un protocole a des formes valides courte et longue d'identification, les receveurs PEUVENT choisir de reconnaître aussi la forme longue.

(Des documents futurs pourront spécifier le contenu du champ Identifiant de source. Cela ne sera pertinent que pour les mises en œuvre qui envoient des paquets encapsulés de type n° 2, car elles sont les seules entités qui ont besoin d'être concernées par la détection des paquets réfléchis de type n° 2.)

5.5.3 Exemple de type n° 1

Un paquet IPv4 (pleinement identifié par un Ethertype de 0x800, exigreant donc le codage de type de protocole de "forme courte") serait transmis avec :

[0xAA-AA-03][0x00-00-5E][0x00-01][pkt$cmi][0x800][paquet IPv4]

Les codets LLC/SNAP différents pour la transmission de paquet en envoi individuel et en diffusion groupée permettent qu'une seule interface IPv4/ATM prenne en charge les deux en démultiplexant sur l'en-tête LLC/SNAP.

6. MARS en détails

La Section 5 a beaucoup d'implications sur le comportement de base du MARS tel qu'observé par les membres de la grappe. La présente section résume le comportement du MARS pour les groupes qui sont fondés sur le maillage de VC, et décrit comment change le comportement d'un MARS lorsque un MCS est enregistré pour la prise en charge d'un groupe.

Le MARS est destiné à être une entité multi protocoles – tous ses tableaux de transposition, les CMI, et les VC de contrôle DOIVENT être gérés dans le contexte du champ mar$pro dans les messages de MARS entrants. Par exemple, un MARS prend complètement en charge les différents ClusterControlVC pour chaque protocole de couche 3 pour lesquels il enregistre des membres. Si un MARS reçoit des messages avec un mar$pro qu'il ne prend pas en charge, le message est abandonné.

En général, le MARS traite les adresses de protocole comme des chaînes d'octets arbitraires. Par exemple, le MARS ne va pas appliquer des vérifications de "classe" spécifiques d'IPv4 aux adresses fournies sous mar$pro = 0x800. Il suffit au MARS de simplement supposer que les points d'extrémité savent comment interpréter les adresses de protocole pour lesquelles ils établissent et libèrent les transpositions.

Le MARS a besoin de messages de contrôle pour porter l'identité du générateur dans le ou les champs d'adresse ATM de source. Les messages qui arrivent avec un champ Numéro ATM vide sont éliminés en silence avant tout autre traitement par le MARS. (Seul le champ Numéro ATM doit être vérifié. Un champ Numéro ATM vide combiné avec un champ Sous adresse ATM non vide ne représente pas une adresse ATM valide.)

(On trouvera en appendice F des exemples de pseudo-code pour un MARS.)

6.1 Interface de base avec les membres de la grappe

Les messages de MARS suivants sont utilisés ou exigés par les membres de la grappe :

1 MARS_REQUEST
2 MARS_MULTI
4 MARS_JOIN
5 MARS_LEAVE
6 MARS_NAK
10 MARS_GROUPLIST_REQUEST
11 MARS_GROUPLIST_REPLY
12 MARS_REDIRECT_MAP

6.1.1 Réponse à MARS_REQUEST

Sauf comme décrit au paragraphe 6.2, si une MARS_REQUEST arrive dont l'adresse ATM de source ne correspond pas à celle d'un membre de la grappe enregistre, le message DOIT être abandonné et ignoré.

6.1.2 Réponse à MARS_JOIN et MARS_LEAVE

Lorsque arrive un enregistrement MARS_JOIN (décrit au paragraphe 5.2.3) le MARS effectue les actions suivantes :
- il ajoute le nœud au ClusterControlVC.
- il alloue un nouvel identifiant de membre de grappe (CMI).
- il insère le nouveau CMI dans le champ mar$cmi du MARS_JOIN.
- il retransmet directement au nœud le MARS_JOIN.

Si le nœud est déjà un membre enregistré de la grappe associée au type spécifié de protocole, son CMI existant est simplement copié dans le MARS_JOIN, et celui-ci est retransmis au nœud. Un seul nœud peut s'enregistrer plusieurs fois si il prend en charge plusieurs protocoles de couche 3. Les CMI alloués par le MARS pour chacun de ces enregistrements peuvent être ou non les mêmes.

L'enregistrement MARS_JOIN retransmis NE DOIT PAS être envoyé sur le ClusterControlVC. Si un membre de la grappe produit un MARS_LEAVE de désenregistrement il est lui aussi retransmis en privé.

Les messages MARS_JOIN et MARS_LEAVE qui ne sont pas d'enregistrement sont ignorés si ils arrivent d'un nœud qui n'est pas enregistre comme membre de grappe.

Les messages MARS_JOIN ou MARS_LEAVE DOIVENT arriver au MARS avec mar$flags.copy réglé à 0, autrement, le message est ignoré en silence.

Tous les messages MARS_JOIN ou MARS_LEAVE sortants DEVRONT avoir mar$flags.copy réglé à 1, et mar$msn réglé au numéro de séquence de grappe pour ClusterControlVC (paragraphe 5.1.4.2).

mar$flags.layer3grp (paragraphe 5.3) DOIT être traité comme remise à zéro pour les MARS_JOIN qui spécifient une seule paire <min,max> couvrant plus d'un seul groupe. Si un MARS_JOIN/LEAVE est reçu qui contient plus d'une paire <min,max>, le MARS DOIT abandonner le message en silence.

Si un ou plusieurs MCS se sont enregistrés auprès du MARS, le traitement de message continue comme décrit au paragraphe 6.2.4.

La base de données du MARS est mise à jour pour ajouter le nœud à tous groupes indiqués dont il n'était pas encore considéré être membre, et le traitement de message se continue comme suit :

Si un seul groupe est joint ou quitté : mar$flags.punched est réglé à 0.

Si le nœud qui se joint (quitte) était déjà (est encore) considéré comme membre du groupe spécifié, le message est retransmis en privé au membre de la grappe. Autrement, le message est retransmis sur ClusterControlVC.

Si un seul bloc couvrant deux groupes ou plus est rejoint ou quitté :
Il est fait une copie du MARS_JOIN/LEAVE original. Cette copie a alors son bloc <min,max> remplacé par un ensemble "à trous" de zéro ou plusieurs paires <min,max>. L'ensemble "à trous" de paires <min,max> couvre la gamme d'adresses entière spécifiée par la paire <min,max> originale, mais exclut les adresses/groupes dont le nœud qui se joint (qui quitte) est déjà (encore) membre du fait d'une adhésion précédente à un seul groupe.

Si aucun "trou" n'a été percé dans le bloc spécifié, le MARS_JOIN/LEAVE original est retransmis sur ClusterControlVC. Autrement, il survient ce qui suit :

Le MARS_JOIN/LEAVE original est retransmis inchangé au membre de grappe source, en utilisant le VC sur lequel il est arrivé. Le champ mar$flags.punched DOIT être remis à 0 dans ce message.

Si l'ensemble "à trous" contient une ou plusieurs paires <min,max>, la copie du MARS_JOIN/LEAVE original est transmise sur le ClusterControlVC, portant la nouvelle liste de <min,max>. Le champ mar$flags.punched DOIT être réglé à 1 dans ce message. (Le champ mar$flags.punched est établi pour assurer que la copie "à trous" est ignorée par la source du message lorsque elle essaye de faire correspondre les messages MARS_JOIN/LEAVE reçus avec ceux envoyés prédédemment (paragraphe 5.2.2)).

Si le MARS reçoit un MARS_LEAVE de désenregistrement (décrit au paragraphe 5.2.3) l'adresse ATM de ce membre DOIT être retirée de tous les groupes auxquels il a pu se joindre, il DOIT sortir du ClusterControlVC, et le CMI DOIT être libéré.

Si le MARS reçoit une ERR_L_RELEASE sur le ClusterControlVC indiquant qu'un membre de grappe s'est déconnecté, l'adresse ATM de ce membre DOIT être retirée de tous les groupes auxquels il a pu se joindre, et le CMI DOIT être libéré.

6.1.3 Génération de MARS_REDIRECT_MAP

Un message MARS_REDIRECT_MAP (décrit au paragraphe 5.4.3) DOIT être transmis sur le ClusterControlVC. Il est RECOMMANDÉ que cela survienne toutes les 1 minute, et cela DOIT survenir au moins toutes les deux minutes. Si le

MARS n'a pas connaissance d'autres MARS de sauvegarde desservant la grappe, il DOIT inclure sa propre adresse comme seule entrée dans le message MARS_REDIRECT_MAP (en plus de remplir les chanps d'adresse de source).

La conception et l'utilisation des entités de MARS de sauvegarde sort du domaine d'application du présent document, et sera traité dans des travaux futurs.

6.1.4 Numéro de séquence de grappe

Le numéro de séquence de grappe (CSN) est décrit au paragraphe 5.1.4, et est porté dans le champ mar$msn des messages de MARS qui sont envoyés aux membres de la grappe (soit sur le ClusterControlVC, soit sur un VC individuel). Le MARS incrémente le CSN après chaque transmission d'un message sur le ClusterControlVC. Le CSN en cours est copié dans le champ mar$msn des messages de MARS qui sont envoyés aux membres de la grappe, sur le ClusterControlVC ou sur un VC privé.

Un MARS devrait être conçu avec soin pour minimiser la possibilité que le CSN ne fasse des bonds sans nécessité. En fonctionnement normal, seuls les membres de la grappe affectés par des problèmes transitoires de liaison vont manquer les mises à jour de CSN et seront forcés de se revalider. Si le MARS lui-même a des à coups, il va être inondé de demandes pendant un temps car tous les membres de la grappe vont tenter de se revalider.

Les calculs sur le CSN DOIVENT être effectués par une arithmétique à 32 bits non signés.

Une implication de ce mécanisme est que le MARS devrait faire en série son traitement de messages MARS_REQUEST, MARS_JOIN et MARS_LEAVE "simultanés". Les opérations d'adhésion et de départ devraient être mises en file d'attente au sein du MARS avec les MARS_REQUEST, et n'être pas traitées jusqu'à ce que tous les paquets de réponse d'une MARS_REQUEST précédente aient été transmis. La transmission de MARS_REDIRECT_MAP devrait être aussi mise en file d'attente de façon similaire.

(La transmission régulière des MARS_REDIRECT_MAP sert également l'objectif secondaire de permettre aux membres de la grappe de suivre les CSN, même si ils ont manqué un MARS_JOIN ou MARS_LEAVE antérieur.)

6.2 Interface de MARS aux serveurs de diffusion groupée (MCS)

Lorsque le MARS retourne les adresses réelles des membres du groupe, le comportement de point d'extrémité décrit à la section 5 résulte en ce que que tous les groupes sont pris en charge par des maillages de VC en point à multipoint. Cependant, lorsque les MCS s'enregistrent pour prendre en charge des groupes de diffusion groupée de couche 3 particuliers, le MARS modifie son utilisation des divers messages de MARS pour amener les points d'extrémité à utiliser à la place le MCS.

Les messages de MARS suivants sont associés aux interactions entr le MARS et les MCS.

3 MARS_MSERV
7 MARS_UNSERV
8 MARS_SJOIN
9 MARS_SLEAVE

Les messages de MARS suivants sont traités d'une manière légèrement différente lorsque les MCS se sont enregistrés pour prendre en charge certaines adresses de groupe :

1 MARS_REQUEST
4 MARS_JOIN
5 MARS_LEAVE

Un MARS doit conserver deux ensembles de transpositions pour chaque groupe de couche 3 en utilisant la prise en charge de MCS. La transposition originale {adresse de couche 3, ATM.1, ATM.2, ... ATM.n} (qu'on appelle maintenant "transposition d'hôte", bien qu'elle inclue des routeurs) est augmentée par une transposition parallèle {adresse de couche 3, serveur.1, serveur.2, ... serveur.K} (la "transposition de serveur"). On suppose qu'aucune adresse ATM n'apparaît à la fois dans la transposition de serveur et dans celle d'hôte pour le même groupe de diffusion groupée. Normalement, K sera à 1, mais il sera supérieur si plusieurs MCS sont configurés pour prendre en charge un groupe donné.

Le MARS conserve aussi un VC en point à multipoint en sortie vers tout MCS enregistré auprès de lui, appelé ServerControlVC (paragraphe 6.2.3). Cela sert à un rôle analogue à celui de ClusterControlVC, permettant au MARS de mettre à jour les MCS avec les changements d'adhésion aux groupes lorsque ils surviennent. Un MARS DOIT aussi

envoyer ses transmissions régulières de MARS_REDIRECT_MAP à la fois sur le ServerControlVC et sur le ClusterControlVC.

6.2.1 Réponse à MARS_REQUEST si le MCS est enregistré

Lorsque le MARS reçoit une MARS_REQUEST pour une adresse qui a à la fois des transpositions d'hôte et de serveur, il génère une réponse fondée sur l'identité de la source de la demande. Si le demandeur est un membre de la transposition de serveur pour le groupe demandé, le MARS retourne alors le contenu de la transposition d'hôte dans une séquence d'un ou plusieurs MARS_MULTI. Autrement, si la source est un membre de grappe valide, le MARS retourne le contenu de la transposition de serveur dans une séquence d'un ou plusieurs MARS_MULTI. Si la source n'est ni un membre de la grappe, ni un membre de la transposition de serveur pour le groupe, la demande est abandonnée et ignorée.

Les serveurs utilisent la transposition d'hôte pour établir un VC de distribution de base pour le groupe. Les membres de la grappe vont établir des VC multipoints sortants avec les membres de la transposition de serveur du groupe, sans savoir que leurs paquets ne vont pas aller directement aux membres du groupe de diffusion groupée.

6.2.2 Messages MARS_MSERV et MARS_UNSERV

MARS_MSERV et MARS_UNSERV sont identiques au message MARS_JOIN. Un MCS utilise un MARS_MSERV avec une paire <min,max> de <X,X> pour spécifier le groupe X de diffusion groupée qu'il veut prendre en charge. Un seul groupe MARS_UNSERV indique le groupe que le MCS ne veut plus prendre en charge. Le code de fonctionnement pour MARS_MSERV est 3 (en décimal), et pour MARS_UNSERV c'est 7 (en décimal).

Ces deux messages sont envoyés au MARS sur un VC en point à point (entre le MCS et le MARS). Après le traitement, ils sont retransmis sur le ServerControlVC pour permettre aux autres MCS de noter le nouveau nœud.

Lors de l'enregistrement ou du désenregistrement de la prise en charge de groupes spécifiques, le fanion mar$flags.register DOIT être à zéro. (Ce fanion n'est mis à un que lorsque le MCS s'enregistre comme membre du ServerControlVC, comme décrit au paragraphe 6.2.3.)

Lorsque un MCS produit un MARS_MSERV pour un groupe spécifique, le message DOIT être abandonné et ignoré si la source ne s'est pas déjà enregistrée auprès du MARS comme serveur de diffusion groupée (paragraphe 6.2.3). Autrement, le MARS ajoute la nouvelle adresse ATM à la transposition de serveur pour le groupe spécifié, en construisant éventuellement une nouvelle transposition de serveur si c'est le premier MCS pour le groupe.

Si un MARS_MSERV représente le premier MCS à s'enregistrer pour un groupe particulier, et si il existe une transposition d'hôte non nulle qui dessert ce groupe particulier, le MARS produit un MARS_MIGRATE (paragraphe 5.1.6) sur le ClusterControlVC. La propre identité du MARS est placée dans les champs Protocole de source et Adresse de matériel du MARS_MIGRATE. L'adresse ATM du MCS est placée comme première et seule adresse ATM cible. L'adresse du groupe affecté est placée dans le champ Adresse cible du groupe de diffusion groupée.

Si un MARS_MSERV n'est pas le premier MCS à s'enregistrer pour un groupe particulier, le MARS change simplement son code de fonctionnement en MARS_JOIN, et envoie une copie du message sur le ClusterControlVC. Cela amène les membres de la grappe à penser qu'un nouveau nœud d'extrémité a été ajouté au groupe spécifié. Dans le MARS_JOIN retransmis le champ mar$flags.layer3grp DOIT être zéro, mar$flags.copy DOIT être un, et mar$flags.register DOIT être zéro.

Lorsque un MCS produit un MARS_UNSERV, le MARS retire son adresse ATM des transpositions de serveur pour tous les groupes spécifiés, supprimant toutes les transpositions de serveur qui se retrouvent être nulles après l'opération.

Le code de fonctionnement est alors changé en MARS_LEAVE et le MARS envoie une copie du message sur le ClusterControlVC. Cela amène les membres de la grappe à penser qu'un nœud d'extrémité a été abandonné dans le groupe spécifié. Dans le MARS_LEAVE retransmis, le champ mar$flags.layer3grp DOIT être zéro, mar$flags.copy DOIT être un, et mar$flags.register DOIT être zéro.

Le MARS retransmet les messages MARS_MSERV et MARS_UNSERV redondants directement au MCS qui les a généré. Les messages MARS_MIGRATE ne sont jamais répétés en réponse aux MARS_MSERV redondants.

Le dernier, ou le seul, MCS pour un groupe PEUT choisir de produire un MARS_UNSERV alors que le groupe a encore des membres. Lorsque le MARS_UNSERV est traité par le MARS, la "transposition de serveur" va être supprimée. Lorsque le MARS_LEAVE associé est produit sur le ClusterControlVC, tous les membres de la grappe qui ont un VC

ouvert avec le MCS pour ce groupe vont le fermer (conformément au paragraphe 5.1.4, car le MCS était leur seul nœud d'extrémité). Lorsque les membres de la grappe trouvent ensuite qu'ils ont besoin de transmettre des paquets au groupe, ils recommencent avec la séquence MARS_REQUEST/MARS_MULTI à établir un nouveau VC. Comme le MARS aura supprimé la transposition de serveur, il va en résulter que la transposition d'hôte sera retournée, et le groupe reviendra à une prise en charge par un maillage de VC.

Le processus inverse est réalisé au moyen du message MARS_MIGRATE lorsque le premier MCS s'enregistre pour prendre un groupe en charge. Cela assure que les membres de la grappe démantèlent explicitement tout maillage de VC qu'ils auraient pu avoir, et rétablissent leur chemin de transmission de diffusion groupée avec le MCS comme étant son point de terminaison.

6.2.3 Enregistrement d'un serveur de diffusion groupée (MCS)

Le paragraphe 5.2.3 décrit comment les points d'extrémité s'enregistrent comme membres de la grappe, et donc sont ajoutés comme nœuds d'extrémité à ClusterControlVC. La même approche est utilisée pour enregistrer les points d'extrémité qui ont l'intention de fournir la prise en charge de MCS.

L'enregistrement auprès du MARS survient lorsque un point d'extrémité produit un MARS_MSERV avec mar$flags.register réglé à un. Après l'enregistrement, le point d'extrémité est ajouté comme nœud d'extrémité à ServerControlVC, et le message MARS_MSERV est retourné en privé au MCS.

Le MCS retransmet ce MARS_MSERV jusqu'à ce qu'il soit confirmé que le MARS l'a reçu (en recevant une copie en retour, de façon analogue au mécanisme décrit au paragraphe 5.2.2 pour la transmission fiable des MARS_JOIN).

Le champ mar$cmi dans MARS_MSERVs DOIT être réglé à zéro à la fois par le MCS et le MARS.

Un MCS peut aussi choisir de se désenregistrer, en utilisant un MARS_UNSERV avec mar$flags.register mis à un. Lorsque cela arrive, le MARS DOIT retirer toutes les références à ce MCS dans toutes les transpositions de serveur associées au protocole (mar$pro) spécifié dans le MARS_UNSERV, et abandonner le MCS dans le ServerControlVC.

Noter que plusieurs MCS logiques peuvent partager la même interface physique ATM, pourvu que chaque MCS utilise une adresse ATM distincte (par exemple, un champ SEL différent dans l'adresse de format NSAP). En fait, un MCS peut partager l'interface ATM d'un nœud qui est aussi un membre de la grappe (comme hôte ou comme routeur) pourvu que chaque entité logique ait une adresse ATM différente.

Un MARS DOIT être capable de traiter une transposition de serveur multi entrées. Cependant, l'utilisation possible de plusieurs MCS s'enregistrant pour prendre en charge le même groupe sera étudiée ultérieurement. En l'absence d'un protocole de synchronisation de MCS, un administrateur de système NE DOIT PAS permettre que plus d'un MCS logique s'enregistre pour un groupe donné.

6.2.4 Réponse modifiée à MARS_JOIN et MARS_LEAVE.

L'existence de MCS qui prennent en charge certains groupes mais pas d'autres exige que le MARS modifie sa distribution de mises à jour seules et en bloc de jonctions/départs aux membres de la grappe. Le MARS ajoute aussi deux nouveaux messages – MARS_SJOIN et MARS_SLEAVE – pour communiquer les changements du groupe aux MCS sur ServerControlVC.

Les messages MARS_SJOIN et MARS_SLEAVE sont identiques à MARS_JOIN, avec respectivement les codes de fonctionnement 18 et 19 (en décimal).

Lorsque un membre de la grappe produit un MARS_JOIN ou MARS_LEAVE pour un seul groupe, le MARS vérifie pour voir si le groupe a une transposition de serveur associée. Si le groupe spécifié n'a pas de transposition de serveur, le traitement continue comme décrit au paragraphe 6.1.2.

Cependant, si il existe une transposition de serveur pour le groupe, un nouvel ensemble d'actions a lieu.

Si le nœud qui se joint (qui quitte) était déjà (n'est plus) considéré comme membre du groupe spécifié, une copie du MARS_JOIN/LEAVE est faite avec le type MARS_SJOIN ou MARS_SLEAVE approprié, et il est transmis sur le ServerControlVC. Cela permet aux MCS qui prennent en charge le groupe de noter le nouveau membre et de mettre à jour leurs VC de données.

Le message original est retransmis inchangé au membre de la grappe de source, en utilisant le VC sur lequel il est arrivé plutôt que ClusterControlVC. Le champ mar$flags.punched DOIT être remis à 0 dans ce message.

(Le paragraphe 5.2.2 exige des membres de la grappe qu'ils aient un mécanisme pour confirmer la réception de leur message par le MARS. Pour les groupes pris en charge par un maillage, l'utilisation de ClusterControlVC sert au double objectif de fournir cette confirmation et de distribuer les informations de mise à jour de groupe. Lorsque un groupe est pris en charge par un MCS, il n'y a pas de raison que tous les membres de la grappe traitent les messages de jonction/départ nuls sur ClusterControlVC, aussi sont-ils renvoyés sur le VC privé entre membre de grappe et MARS.)

La réception d'un bloc MARS_JOIN (par exemple, provenant d'un routeur qui arrive en ligne) ou d'un MARS_LEAVE, exige une réponse plus complexe. Le seul bloc <min,max> peut simultanément couvrir des groupes pris en charge par un maillage et des groupes pris en charge par MCS. Cependant, les membres de la grappe ont seulement besoin d'être informés des groupes pris en charge par maillage auxquels s'est joint le point d'extrémité. Tout ce que les MCS ont besoin de savoir est si le point d'extrémité se joint à des groupes pris en charge par MCS.

La solution est de modifier le MARS_JOIN ou MARS_LEAVE qui est retransmis sur ClusterControlVC. On effectue l'action suivante : on fait une copie du MARS_JOIN/LEAVE avec le type MARS_SJOIN ou MARS_SLEAVE selon ce qui est approprié, avec son bloc <min,max> remplacé par un ensemble "à trous" de zéro, une ou plusieurs paires <min,max>. L'ensemble "à trous" de paires <min,max> couvre la gamme entière des adresses spécifiées par la paire <min,max> originale, mais exclut les adresses/groupes dont le nœud qui se joint (qui quitte) est déjà (est encore) membre du fait d'une adhésion individuelle antérieure au groupe.

Avant transmission sur le ClusterControlVC, le MARS_JOIN/LEAVE d'origine a alors son bloc <min,max> remplacé par un ensemble "à trous" de zéro, une ou plusieurs paires <min,max>. L'ensemble "à trous" de paires <min,max> couvre la gamme entière des adresses spécifiées par la paire <min,max> originale, mais exclut les adresses/groupes pris en charge par les MCS ou dont le nœud qui se joint (qui quitte) est déjà (est encore) membre du fait d'une adhésion individuelle antérieure.

Si aucun "trou" n'a été percé dans le bloc spécifié, le MARS_JOIN/LEAVE d'origine est retransmis inchangé sur ClusterControlVC. Autrement, il arrive ce qui suit :

> Le MARS_JOIN/LEAVE d'origine est retransmis inchangé au membre de la grappe source, en utilisant le VC sur lequel il est arrivé. Le champ mar$flags.punched DOIT être remis à 0 dans ce message.
>
> Si l'ensemble à trous contient une ou plusieurs paires <min,max>, une copie du MARS_JOIN/LEAVE original est transmise sur le ClusterControlVC, portant la nouvelle liste des <min,max>. Le champ mar$flags.punched DOIT être mis à 1 dans ce message.
>
> Le champ mar$flags.punched est réglé de façon à assurer que la copie à trous sera ignorée par la source du message quand elle essayera de faire correspondre les messages MARS_JOIN/LEAVE reçus avec ceux envoyés précédemment (paragraphe 5.2.2).

(L'appendice A discute de certains algorithmes pour le "perçage de trous".)

On suppose que les MCS utilisent les MARS_SJOIN et les MARS_SLEAVE pour mettre à jour leurs propres VC avec les membres réels du groupe.

Le mar$flags.layer3grp recopié dans les messages transmis par le MARS. mar$flags.copy DOIT être mis à 1.

6.2.5 Numéros de séquence pour le trafic de ServerControlVC

Comme pour le numéro de séquence de grappe, le MARS tient un numéro de séquence de serveur (SSN) qui est incrémenté après chaque transmission sur le ServerControlVC. La valeur courante du SSN est insérée dans le champ mar$msn de chaque message que le MARS produit dont il pense qu'il est destiné à un MCS. Cela inclut les MARS_MULTI qui sont retournés en réponse à une MARS_REQUEST provenant d'un MCS, et d'un MARS_REDIRECT_MAP envoyé sur le ServerControlVC. Le MARS doit vérifier la source des MARS_REQUEST, et si elle est un MCS enregistré, le SSN est copié dans le champ mar$msn, autrement, le CSN est copié dans le champ mar$msn.

Les MCS sont supposés garder la trace et utiliser les SSN d'une façon analogue à celle dont les points d'extrémité utilisent le CSN au paragraphe 5.1 (pour déclancher la revalidation des informations d'adhésion aux groupes).

Un MARS devrait être conçu avec soin pour minimiser la possibilité que le SSN fasse des sauts inutiles. En fonctionnement normal, seuls les MCS qui sont affectés par des problèmes transitoires de liaisin vont manquer des mises à

jour de mar$msn et seront forcés de se revalider. Si le MARS lui-même a un fonctionnement chaotique, il va être inondé de demandes pendant un temps car tous les MCS vont tenter de se revalider.

6.3 Pourquoi des numéros de séquence globaux ?

Le CSN et le SSN sont globaux dans le contexte d'un protocole donné (par exemple, IPv4, mar$pro = 0x800). Ils comptent l'activité du ClusterControlVC et du ServerControlVC sans référence au ou aux groupes de diffusion groupée impliqués. Cela peut être perçu comme une limitation, parce qu'il n'y a pas de moyen pour les membres de la grappe ou pour les serveurs de diffusion groupée d'isoler exactement pour quel groupe de diffusion groupée ils peuvent avoir manqué une mise à jour. Une solution de remplacement était d'essayer de fournir un numéro de séquence par groupe.

Malheureusement, les numéros de séquence par groupe ne sont pas praticables. Le mécanisme actuel permet que les informations de séquence soient portées sur les messages de MARS qui sont déjà en transit pour d'autres raisons. La capacité à spécifier des blocs d'adresses de diffusion groupée avec un seul MARS_JOIN ou MARS_LEAVE signifie qu'un seul message peut se référer à un changement de l'adhésion simultanément pour plusieurs groupes. Un seul champ mar$msn ne peut pas fournir des informations significatives sur la séquence de chaque groupe. Plusieurs champs mar$msn aurait été trop lourd.

Tout MARS ou membre de grappe qui prend en charge différents protocoles DOIT conserver des tableaux de transposition et des numéros de séquence séparés pour chaque protocole.

6.4 Architectures MARS redondantes/de sauvegarde

Si il existe un MARSs de sauvegarde pour une grappe donnée, des mécanismes sont alors nécessaires pour assurer la cohérence entre leurs tableaux de transposition et ceux du MARS actif en cours.

(Les membres de la grappe vont considérer que les MARS de sauvegarde existent si ils ont été configurés avec un tableau des adresses des MARS, ou si les messages MARS_REDIRECT_MAP réguliers contiennent une liste de deux adresses ou plus.)

La définition d'un protocole de synchronisation de MARS sort du domaine d'appication du présent document, et fera vraisemblablement l'objet de travaux de recherche complémentaires. Cependant, on peut faire les observations suivantes :

Les messages MARS_REDIRECT_MAP existent pour permettre à un MARS de forcer les points d’extrémité à passer à un autre MARS (par exemple, dans les répercussions d'une défaillance d'un MARS; le MARS de sauvegarde choisi va éventuellement souhaiter passer le contrôle de la grappe au MARS principal lorsque il est revenu en fonctionnement normal).

Les membres de la grappe et les MCS n'ont pas besoin de démarrer avec la connaissance de plus d'un MARS, pourvu que le MARS produise correctement les messages MARS_REDIRECT_MAP avec la liste complète des MARS pour cette grappe.

Tous les mécanismes pour la synchronisation des MARS de sauvegarde (et ceux qui prennent en compte la récupération des défaillances de MARS) devraient être compatibles avec le comportement de membre de grappe décrit dans ce document.

7. Comment un MCS utilise un MARS

Lorsque un MCS prend en charge un groupe de diffusion groupée, il agit comme un mandataire de point d’extrémité de grappe pour les envoyeurs au groupe. Il se comporte aussi de façon analogue à un envoyeur, gérant un seul VC sortant en point à multipoint avec les membres réels du groupe.

Une description détaillée des architectures possibles de MCS sort du domaine d'application du présent document. Cette section va souligner les questions principales.

7.1 Association avec un groupe de couche 3 particulier

Lorsque un MCS produit un MARS_MSERV, il force tous les envoyeurs au groupe de couche 3 spécifié à terminer leurs VC sur l'adresse ATM de source fournie.

La plus simple architecture de MCS implique de prendre les AAL_SDU entrants et de simplement les renvoyer sur un seul VC en point à multipoint. Un tel MCS ne peut pas prendre en charge plus d'un groupe à la fois, car il n'a aucun moyen de différencier le trafic destiné aux différents groupes. À utiliser cette architecture, un nœud physique fournirait la prise en charge du MCS pour plusieurs groupes en créant plusieurs instances logiques de MCS, chacune avec une adresse ATM différente (par exemple, une valeur de SEL différente dans le NSAPA du nœud).

Une approche légèrement plus complexe serait d'ajouter un traitement minimal spécifique de couche 3 dans le MCS. Elle regarderait à l'intérieur des AAL_SDU reçus et déterminerait à quel groupe de couche 3 ils sont destinés. Une seule instance d'un tel MCS pourrait enregistrer son adresse ATM auprès du MARS pour plusieurs groupe de couche 3, et gérer plusieurs VC point à multipoint sortants indépendants (un pour chaque groupe).

Lorsque démarre un MCS, il DOIT s'enregistrer auprès du MARS, comme décrit au paragraphe 6.2.3, en identifiant les protocoles qu'il prend en charge avec le champ mar$pro du message MARS_MSERV. Ceci s'applique aussi aux MCS logiques, même si ils partagent la même interface ATM physique. C'est important afin que le MARS puisse réagir à la perte d'n MCS lorsque il abandonne le ServerControlVC. (Une conséquence est que les architectures de "simple" MCS se terminent avec un membre ServerControlVC par groupe. Les MCS avec un traitement spécifique de couche 3 peuvent prendre en charge plusieurs groupes tout en ne s'enregistrant que comme un seul membre du ServerControlVC.)

Un MCS NE DOIT PAS partager la même adresse ATM qu'un membre de la grappe, bien qu'il puisse partager la même interface ATM physique.

7.2 Terminaison des VC entrants

Un MCS DOIT terminer les VC unidirectionnels de la même manière qu'un membre de la grappe. (Par exemple, terminer sur une entité de LLC lorsque l'encapsulation LLC/SNAP est utilisée, comme décrit dans la RFC1755 pour les points d’extrémité en envoi individuel.)

7.3 Gestion d’un VC sortant

Un MCS DOIT établir et gérer son VC de point à multipoint sortant comme le fait un membre de la grappe (paragraphe 5.1).

MARS_REQUEST est utilisé par le MCS pour établir les nœuds d'extrémité initiaux pour le VC point à multipoint sortant du MCS. Après l'établissement du VC, le MCS réagit aux MARS_SJOIN et aux MARS_SLEAVE de la même façon qu'un membre de la grappe réagit aux MARS_JOIN et aux MARS_LEAVE.

Le MCS garde trace du numéro de séquence de serveur des champs mar$msn des messages provenant du MARS, et revalide son ou ses VC point à multipoint sortants lorsque un saut de numéro de séquence survient.

7.4 Utilisation d’un MARS de sauvegarde

Le MCS utilise la même approche pour les MARS de sauvegarde que celle d'un membre de la grappe (paragraphe 5.4), gardant trace des messages MARS_REDIRECT_MAP sur le ServerControlVC.

8. Prise en charge des routeurs de diffusion groupée IP

Les routeurs de diffusion groupée sont nécessaires pour la propagation du trafic de diffusion groupée par delà les contraintes d'une seule grappe (trafic inter grappe). (En un sens, il y a des serveurs de diffusion groupée qui agissent à la couche immédiatement supérieure, avec des grappes plutôt que des points d’extrémité individuels, comme source et destination abstraites.)

Les routeurs de diffusion groupée participent normalement à des algorithmes et politiques d'acheminement de diffusion groupée de couche supérieure qui sortent du domaine d'application du présent mémoire (par exemple, DVMRP [5] dans l'environnement IPv4).

On suppose que les routeurs de diffusion groupée seront mis en œuvre sur la même sorte d'interface ATM/IP qu'utiliserait un hôte de diffusion groupée. Leurs interfaces ATM/IP vont s'enregistrer auprès du MARS comme les membres de la grappe, se joignant et quittant en tant que de besoin les groupes de diffusion groupée. Comme noté à la section 5, plusieurs

"points d'extrémité" logiques peuvent être mis en œuvre sur une seule interface ATM physique. Les routeurs utilisent cette approche pour fournir des interfaces dans chacune des grappes entre lesquelles ils vont acheminer.

Le reste de cette section supposera un scénario IPv4 simple dans lequel la portée d"une grappe a été limitée à un LIS particulier qui fait partie d'un réseau IP sous-jacent. Tous les membres du LIS ne sont pas nécessairement des membres enregistrés de la grappe (on peut avoir des hôtes en envoi individuel seulement dans le LIS).

8.1 Transmission dans une grappe

Si le routeur de diffusion groupée a besoin de transmettre un paquet à un groupe au sein de la grappe, son interface ATM/IP va ouvrir un VC de la même manière que le ferait un hôte normal. Une fois qu'un VC est ouvert, le routeur surveille les messages MARS_JOIN et MARS_LEAVE et leur répond comme le ferait un hôte normal.

Le côté émission du routeur de diffusion groupée DOIT mettre en œuvre des temporisateurs d'inactivité pour fermer les VC sortants inactifs comme pour les hôtes normaux.

Comme avec un hôte normal, le routeur de diffusion groupée n'a pas besoin d'être membre d'un groupe auquel il envoie.

8.2 Adhésion en mode "disparate"

Une fois enregistré et initialisé, le plus simple modèle de fonctionnement de routeur de diffusion groupée IPv4 est de produire un MARS_JOIN qui renferme l'espace des adresses de classe D tout entier. En effet il devient "disparate", car il sera un nœud d'extrémité pour tous les VC multipoints présents et futurs établis pour les groupes IPv4 sur la grappe.

La question de la façon dont on choisit à quels groupes propager en dehors de la grappe sort du domaine d'application de ce document.

Conformément à la RFC1112, les routeurs de diffusion groupée IP peuvent conserver l'utilisation des messages d'interrogation IGMP et de rapport IGMP pour s'assurer de l'appartenance des membres au groupe. Cependant, certaines optimisations sont possibles, et sont décrites au paragraphe 8.5.

8.3 Transmission dans la grappe

Dans certaines circonstances, la grappe peut simplement être à un autre bond entre les sous-réseaux IP qui ont des participants à un groupe de diffusion groupée.

[LAN.1] ----- IPmcR.1 -- [grappe/LIS] -- IPmcR.2 ----- [LAN.2]

LAN.1 et LAN.2 sont des sous-réseaux (comme des Ethernet) avec des hôtes rattachés qui sont membres du groupe X.

IPmcR.1 et IPmcR.2 sont des routeurs de diffusion groupée qui ont des interfaces avec le LIS.

Une solution traditionnelle serait de traiter le LIS comme un sous-réseau en envoi individuel, et d'utiliser des routeurs de tunnelage. Cependant, cela ne permettrait pas aux hôtes sur le LIS de participer au trafic trans-LIS.

Supposons que IPmcR.1 reçoive de façon disparate des paquets sur son interface LAN.1. Supposons de plus qu'il est configuré pour propager du trafic de diffusion groupée à toutes les interfaces rattachées. Dans ce cas, cela signifie le LIS.

Lorsque un paquet pour le groupe X arrive sur son interface LAN.1, IPmcR.1 envoie simplemendt le paquet au groupe X sur l'interface de LIS comme le ferait un hôte normal (émettant une MARS_REQUEST pour le groupe X, créant le VC, envoyant le paquet).

En supposant que IPmcR.2 s'est initialisé auprès du MARS comme membre de l'espace de classe D entier, il aura été retourné comme membre de X même si aucun autre nœud n'est membre sur le LIS. Tous les paquets pour le groupe X reçus sur l'interface de LIS de IPmcR.2 peuvent être retransmis sur le LAN.2.

Si IPmcR.1 est initialisé de façon similaire, le processus inverse va s'appliquer pour le trafic de diffusion groupée de LAN.2 à LAN.1, pour tout groupe de diffusion groupée. L'avantage de ce scénario est que les membres de la grappe au sein du LIS peuvent aussi se joindre au groupe X et le quitter à tout moment.

8.4 Adhésion en mode "semi disparate"

Les routeurs IP d'envoi individuel et de diffusion groupée ont un problème commun – les limitations sur le nombre de contextes AAL disponibles à leurs interfaces ATM. Être "disparate" au sens de la [RFC1112] signifie que pour chaque M hôtes qui envoient à N groupes, l'interface ATM d'un routeur de diffusion groupée aura M*N moteurs de réassemblage entrants engagés.

Il n'est pas difficile d'imaginer des situations où un certain nombre de groupes de diffusion groupée sont actifs au sein du LIS mais il n'est pas obligé qu'ils soient propagés au-delà du LIS lui-même. Un exemple pourrait être un système de simulation réparti spécifiquement conçu pour utiliser l'environnement IP/ATM à haut débit. Il pourrait n'y avoir aucun moyen pratique d'utiliser ce trafic de "l'autre côté" du routeur de diffusion groupée, et donc dans le schéma conventionnel, le routeur devrait de toutes façon être une extrémité pour chaque hôte participant.

Comme ce problème survient en dessous de la couche IP, il vaut de noter que tous les mécanismes de "portée" au niveau de l'acheminement de la diffusion groupée IP ne fournissent pas de solution. Une portée de niveau IP résulterait encore en ce que l'interface ATM du routeur recevrait le trafic sur les groupes à portée définie,juste pour l'éliminer.

Dans cette situation, l'administrateur de réseau peut configurer ses routeurs de diffusion groupée pour exclure les sections de l'espace d'adresses de classe D lorsque il émet des MARS_JOIN. Les groupes de diffusion groupée qui ne seront jamais propagés au-delà de la grappe n'auront pas le routeur sur leur liste des membres, et le routeur n'aura jamais à recevoir (et simplement ignorer) le trafic provenant de ces groupes.

Un autre scénario implique le produit M*N lorsque il excède la capacité de l'interface d'un seul routeur (en particulier si cette même interface doit aussi prendre en charge un service de routeur IP en envoi individuel).

Un administrateur de réseau peut choisir d'ajouter un second nœud, pour fonctionner comme des routeurs de diffusion groupée IP en parallèle. Chaque routeur serait configuré pour être "disparate" sur des parties séparées de l'espace d'adresses de la classe, ne s'exposant donc qu'à une partie de la charge des VC. Ce partage serait complètement transparent aux hôtes IP au sein du LIS.

Le mode disparate restreint ne rompt pas avec l'utilisation des messages de rapport IGMP de la RFC1112. Si le routeur est configuré pour servir un bloc particulier des adresses de classe D, il va recevoir le rapport IGMP. Si le routeur n'est pas configuré pour prendre en charge un bloc particulier, l'existence d'un rapport IGMP pour un groupe dans ce bloc n'est alors pas pertinente pour le routeur. Tous les routeurs sont de toutes façons capables de suivre les changements des adhésions par le trafic de MARS_JOIN et de MARS_LEAVE. (Le paragraphe 8.5 discute d'une meilleure solution de remplacement à IGMP au sein d'une grappe.)

Les mécanismes et les raisons de l'établissement de ces modes de fonctionnement sortent du domaine d'application de ce document.

8.5 Solution de remplacement aux interrogations IGMP

Un aspect malencontreux d'IGMP est qu'il suppose que la diffusion groupée des paquets IP est un événement trivial et peu coûteux à la couche de liaison. Par conséquent, les interrogations IGMP régulières sont en diffusion groupée par les routeurs au groupe 224.0.0.1. Ces interrogations sont destinées à déclancher des réponses IGMP de la part des membres de la grappe qui ont des membres de couche 3 de groupes particuliers.

Les messages MARS_GROUPLIST_REQUEST et MARS_GROUPLIST_REPLY étaient conçus pour permettre aux routeurs d'éviter d'avoir à transmettre en fait des interrogations IGMP en dehors d'une grappe.

Chaque fois que le moteur de transmission du routeur souhaite transmettre une interrogation IGMP, une MARS_GROUPLIST_REQUEST peut être envoyée à la place au MARS. La ou les MARS_GROUPLIST_REPLY résultantes (décrites au paragraphe 5.3) de la part du MARS portent toutes les informations que le routeur aurait obtenues des réponses IGMP.

Il est RECOMMANDÉ que les routeurs de diffusion groupée utilisent ce service de MARS pour minimiser le trafic IGMP au sein de la grappe.

Par défaut, une MARS_GROUPLIST_REQUEST DEVRAIT spécifier l'espace d'adresses entier (par exemple, <224.0.0.0, 239.255.255.255> dans un environnement IPv4). Cependant, les routeurs qui desservent une partie de l'espace d'adresses (comme décrit au paragraphe 8.4) PEUVENT choisir de produire des MARS_GROUPLIST_REQUEST qui ne spécifient que le sous-ensemble de l'espace d'adresses qu'ils desservent.

À première vue , il semblerait aussi utile que les routeurs de diffusion groupée gardent trace des MARS_JOIN et des MARS_LEAVE qui arrivent avec mar$flags.layer3grp établi. Cela pourrait être utilisé an lieu des rapports IGMP, pour fournir à temps au routeur l'indication qu'un nouveau membre du groupe de couche 3 existe au sein de la grappe. Cependant, cela ne fonctionnera que sur les groupes pris en charge par des maillages de VC et n'est donc PAS RECOMMANDÉ).

L'Appendice B expose des mécanismes moins élégants pour réduire l'impact du trfic IGMP au sein d'une grappe, dans l'hypothèse où les interfaces ATM/IP avec la grappe sont utilisées par un code de diffusion groupée non optimisé.

8.6 CMI à travers plusieurs interfaces

L'identifiant de membre de grappe (CMI) n'est unique qu'au sein de la grappe gérée par un MARS particulier. À première vue, cela peut paraître poser un problème lorsque un routeur de diffusion groupée achemine entre deux grappes ou plus en utilisant une seule interface physique ATM. Le routeur va s'enregistrer auprès de deux MARS ou plus, et acquérir par là les CMI correspondants indépendants. Étant donné qu'aucun MARS n'a de raison de synchroniser son allocation de CMI, il est possible qu'un hôte dans une grappe ait le même CMI que l'interface du routeur à une autre grappe. Comment le routeur va t-il distinguer entre ses propres paquets réfléchis, et les paquets de cet autre hôte ?

La réponse tient au fait que les routeurs (et les hôtes) mettent en fait en œuvre des interfaces ATM/IP logiques sur une seule interface physique ATM. Chaque interface logique aura une adresse ATM unique (par exemple, un NSAP avec différents champ SELector, un pour chaque interface logique).

Chaque interface logique ATM/IP est configurée avec l'adresse d'un seul MARS, ne se rattache qu'à une seule grappe, et n'a donc à se soucier que d'un seul CMI. Chacun des MARS auprès desquels le routeur s'est enregistré aura reçu une adresse ATM différente (correspondant aux différentes interfaces logiques ATM/IP) dans chaque enregistrement MARS_JOIN.

Lorsque les hôtes d'une grappe ajoutent le routeur comme nœud d'extrémité, ils vont spécifier l'adresse ATM de l'interface logique ATM/IP appropriée sur le routeur dans le message L_MULTI_ADD. Donc, chaque interface logique ATM/IP n'aura à vérifier et filtrer que sur les CMI alloués par son propre MARS.

Par nature, la différenciation de grappe est réalisée en s'assurant que les interfaces logiques ATM/IP sont affectées à des adresses ATM différentes.

9. Applications multiprotocoles de MARS et de clients MARS

On fait ici une tentative délibérée de décrire le MARS et ses mécanismes associés de façon indépendante d'un protocole de couche supérieure spécifique qui fonctionnerait sur le nuage ATM. L'application immédiate du présent document est l'environnement IPv4, et cela est reflété par le choix des exemples clés. Cependant, les champs mar$pro.type et mar$pro.snap dans tout message de contrôle MARS permettent à tout protocole de couche supérieure, qu'il ait une identification de protocole de "forme courte" ou de "forme longue" (paragraphe 4.3) d'être pris en charge par un MARS.

Tout MARS DOIT mettre en œuvre des tableaux de transposition logique et une prise en charge entièrempent séparés. Tout membre de grappe doit interpréter les messages provenant du MARS dans le contexte du type de protocole auquel le message du MARS se réfère.

Tout MARS et client MARS DOIT traiter les identifiants de membre de grappe dans le contexte du type de protocole porté dans le message de MARS ou le paquet de données qui contient le CMI.

Par exemple, un Ethertype de 0x86DD a été allpué à IPv6. Cela signifie que la "forme courte" d'identification de protocole doit être utilisée dans les messages de contrôle de MARS et l'encapsulation du chemin des données (paragraphe 5.5). Un client de diffusion groupée IPv6 règle le champ mar$pro.type de chaque message de MARS à 0x86DD. Lorsque ils portent des adresses IPv6, les champs mar$spln et mar$tpln sont soit à 0 (pour les informations nulles ou non existantes) soit à 16 (pour l'adresse IPv6 complète).

Suivant les règles du paragraphe 5.5, un paquet de données IPv6 est encapsulé selon :

[0xAA-AA-03][0x00-00-5E][0x00-01][pkt$cmi][0x86DD][paquet IPv6]

Un hôte ou interface de point d’extrémité qui utilise le même MARS pour prendre en charge plusieurs protocoles pour les besoins de la diffusion groupée NE DOIT PAS supposer que leur CMI sera le même pour chaque protocole.

10. Traitement des paramètres supplémentaires

Le champ mar$extoff dans [En-tête fixe] indique si des paramètres supplémentaires sont portés par un message de contrôle MARS. Ce mécanisme est destiné à permettre l'ajout de nouvelles fonctionnalités au protocole MARS dans des documents ultérieurs.

Les paramètres supplémentaires sont convoyés comme une liste d'éléments d'information codés en TLV (type, longueur, valeur). Le TLV commence sur la première frontière de 32 bits qui suit le champ [Adresses] dans le message de contrôle de MARS (par exemple, après mar$tsa.N dans un MARS_MULTI, après mar$max.N dans un MARS_JOIN, etc).

10.1 Interprétation du champ mar$extoff

Si le champ mar$extoff n'est pas à zéro, il indique qu'une liste d'un ou plisieurs TLV est ajoutée au message de MARS. Le premier TLV est trouvé en traitant mar$extoff comme une représentation d'entier non signé d'un décalage (en octets) depuis le début du message MARS (le MSB du champ mar$afn).

Comme les TLV sont verrouillés sur 32 bits, les deux derniers bits de mar$extoff sont aussi réservés. Un receveur DOIT masquer ces deux bits avant de calculer le décalage d'octet de la liste de TLV. Un envoyeur DOIT régler ces deux bits à zéro.

Si mar$extoff est à zéro, aucun TLVn'a été ajouté.

10.2 Format des TLV

Lorsque ils existent, les TLV commencent sur une frontière de 32 bits, sont des multiples de 32 bits, et forment une liste séquentielle terminée par un TLV nul.

La structure du TLV est:

[Type - 2 octets][Longueur - 2 octets][Valeur - n*4 octets]

Le sous champ Type indique comment le contenu du sous champ Valeur doit être interprété.

Le sous champ Longueur indique le nombre d'octets valides dans le sous champ Valeur. Les octets valides dans le sous champ Valeur commencent immédiatement après le sous champ Longueur. Le décalage (en octets) à partir du début de ce TLV jusqu'au début du prochain TLV dans la liste est donné par la formule suivante :

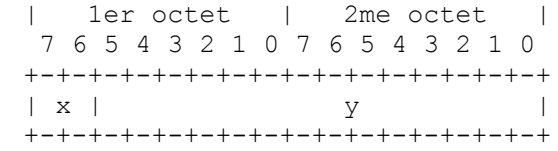
décalage = (longueur + 4 + ((4-(longueur & 3)) % 4))

(où % est l'opérateur modulo)

Le sous champ Valeur est bourré avec 0, 1, 2, ou 3 octets pour garantir que le prochain TLV est verrouillé sur 32 bits. Les localisations objet du bourrage DOIVENT être mises à zéro.

(Par exemple, un TLV qui n'a besoin que de 5 octets d'informations valides sera long de 12 octets. Le sous champ Longueur contiendra la valeur 5, et le sous champ Valeur sera bourré jusqu'à 8 octets. Les 5 octets d'informations valides commencent au premier octet du sous champ Valeur.)

Le sous champ Type est formaté de la façon suivante :



Les deux bits de poids fort (Type.x) déterminent comment devrait se comporter un receveur lorsque il ne reconnaît pas le type de TLV indiqué par les 14 bits de moindre poids (Type.y). Les comportements requis sont :

Type.x = 0 Sauter le TLV, continuer à traiter la liste.
Type.x = 1 Arrêter le traitement, abandonner en silence le message MARS.
Type.x = 2 Arrêter le traitement, abandonnerle message, donner une indication d'erreur.
Type.x = 3 Réservé. (traité en fait comme x = 0)

(L'indication d'erreur générée lorsque Type.x = 2 DEVRAIT être enregistrée dans un journal d'événements d'une façon significative en local. L'activité de messages MARS en réponse à de telles conditions d'erreur sera définie dans des documents futurs.)

L'espace de type de TLV (Type.y) est redivisé pour permettre des utilisation en dehors de celles définies par l'IETF :

0	TLV nul.
0x0001 - 0x0FFF	Réservé pour l'IETF.
0x1000 - 0x11FF	Alloué à l'ATM Forum.
0x1200 - 0x37FF	Réservé pour l'IETF.
0x3800 - 0x3FFF	Utilisation expérimentale.

10.3 Traitement des messages MARS avec des TLV

Les paramètres supplémentaires agissent comme agents de modification du comportement de base spécifié par le champ mar$op de tout message MARS.

Si un message MARS arrive avec un champ mar$extoff non à zéro, sa liste de TLV DOIT être analysée avant le traitement du message MARS conformément à la valeur de mar$op. Les TLV non reconnus DOIVENT être traités comme requis par leur valeur de Type.x.

Comment les TLV modifient les opérations MARS de base sera spécifique de mar$op et du TLV.

10.4 Ensemble initial d'éléments TLV

La conformité au présent document EXIGE seulement la reconnaissance d'un TLV, le TLV nul. Il termine une liste de TLV, et DOIT être présent si mar$extoff est différent de zéro dans un message MARS. Il PEUT être le seul TLV présent.

Le TLV nul est codé par :

[0x00-00][0x00-00]

De futurs documents décriront les formats, contenus, et interprétations de TLV supplémentaires. Les exigences minimales d'analyse imposées par le présent document sont destinées à permettre aux mises en œuvre conformes de MARS et de client MARS de s'adapter en douceur et de façon prévisible aux développements futurs de TLV.

11. Décisions clés et questions ouvertes

Ce document propse les décisions clés suivantes :

On propose un serveur de résolution d'adresse de diffusion groupée (MARS, *Multicast Address Resolution Server*) pour coordonner et répartir les transpositions d'adresses ATM de points d'extrémité en adresses de groupe de diffusion groupée de couche supérieure arbitraire. Le cas spécifique de la diffusion groupée IPv4 est utilisé comme exemple.

Le concept de "grappe" est introduit pour définir la portée de la responsabilité d'un MARS, et l'ensemble des points d'extrémité ATM qui veulent participer à la diffusion groupée de niveau liaison.

Un MARS est décrit avec les fonctionnalités requises pour prendre en charge la diffusion groupée intra grappe en utilisant soit des maillages de VC soit des serveurs de diffusion groupée (MCS) de niveau ATM.

L'encapsulation LLC/SNAP des messages de contrôle MARS permet au trafic de MARS et d'ATMARP de partager les VC, et de permettre des entités MARS et ATMARP partiellement co-résidentes.

Nouveaux types de message :

MARS_JOIN, MARS_LEAVE, MARS_REQUEST. Ils permettent aux points d'extrémité de se joindre, de quitter, et de demander la liste des membres actuels des groupes de diffusion groupée.

MARS_MULTI. Permet que plusieurs adresses ATM soient retournées par les MARS en réponse à un MARS_REQUEST.

MARS_MSERV, MARS_UNSERV. Permet aux serveurs de diffusion groupée de s'enregistrer et se désenregistrer auprès du MARS.

MARS_SJOIN, MARS_SLEAVE. Permet au MARS de faire passer des changements d'adhésion aux serveurs de diffusion groupée.

MARS_GROUPLIST_REQUEST, MARS_GROUPLIST_REPLY. Permet au MARS d'indiquer quels groupes ont actuellement des membres de couche 3. Ils peuvent être utilisés pour la prise en charge de IGMP dans les environnements IPv4, et des fonctions similaires dans d'autres environnements.

MARS_REDIRECT_MAP. Permet au MARS de spécifier un ensemble d'adresses de MARS de sauvegarde.

MARS_MIGRATE. Permet au MARS de forcer les membres de la grappe à passer d'un maillage de VC à une arborescence de transmission fondée sur des MCS en une seule opération.

Les entrées de tableau de transposition de MARS avec des caractères génériques sont possibles, lorsque une seule adresse ATM est simultanément associée à des blocs d'adresses de groupes de diffusion groupée.

Pour le protocole MARS, mar$op.version = 0. L'ensemble complet des messages de contrôle MARS et des valeurs de mar$op.type est :

1	MARS_REQUEST
2	MARS_MULTI
3	MARS_MSERV
4	MARS_JOIN
5	MARS_LEAVE
6	MARS_NAK
7	MARS_UNSERV
8	MARS_SJOIN
9	MARS_SLEAVE
10	MARS_GROUPLIST_REQUEST
11	MARS_GROUPLIST_REPLY
12	MARS_REDIRECT_MAP
13	MARS_MIGRATE

Un certain nombre de questions restent ouvertes au stade actuel et feront vraisemblablement l'objet de recherches et de documents supplémentaires qui complèteront celui-ci.

Le comportement spécifié des points d'extrémité permet l'utilisation de MARS redondants/de sauvegarde au sein d'une grappe. Cependant, aucune spécification n'existe encore sur la façon dont ces MARS se coordonnent entre eux. (Par défaut, on a seulement un MARS par grappe.)

Le comportement spécifié des points d'extrémité et du service MARS permet l'utilisation de plusieurs MCS par groupe. Cependant, aucune spécification n'existe encore sur la façon dont cela peut être utilisé, ou comment ces MCS se coordonnent entre eux. En l'attente de travaux futurs sur des protocoles de coordination de MCS, on a par défaut un seul MCS par groupe.

Le MARS s'appuie sur le fait que le membre de la grappe abandonne le ClusterControlVC si il est mort. On ne sait pas trop si des mécanismes supplémentaire sont nécessaires pour détecter et supprimer les membres de la grappe "morts".

La prise en charge de la "diffusion" de couche 3 comme cas particulier de diffusion groupée (où le "groupe" englobe tous les membres de la grappe) n'a pas été explicitement discutée.

La prise en charge de "l'envoi individuel" de couche 3 comme cas particulier de diffusion groupée (où le "groupe" est un seul membre de grappe, identifié par l'adresse de protocoele d'envoi individuel du membre de grappe) n'a pas été explicitement discutée.

Les futurs développements du groupe ATM "Addresses" et "Leaf Initiated Join" de la spécification UNI au forum ATM n'ont pas été traités. (Cependant, les problèmes identifiés dans le présent document par rapport à la rareté des VC et à l'impact sur les contextes AAL ne seront pas réglés par de tels développements dans le protocole de signalisation.)

Les possibles modifications de l'interprétation des champs mar$hrdrsv et mar$afn dans l'en-tête fixe, fondées sur des valeurs différentes de mar$op.version, feront d'objet d'études complémentaires.

Considérations pour la sécurité

Les questions de sécurité ne sont pas abordées dans le présent document.

Remerciements

Les discussions au sein du groupe de travail IP sur ATM ont aidé à préciser les idées exprimées dans ce document. John Moy (Cascade Communications Corp.) avait initialement suggéré l'idée d'entrées génériques dans le serveur ARP. Drew Perkins (Fore Systems) a fourni une critique rigoureuse et utile des premier mécanismes proposés pour distribuer et valider les informations d'adhésion au groupe. Susan Symington (et ses collègues de MITRE Corp., Don Chirieleison, et Bill Barns) ont clairement articulé la nécessité de la prise en charge par le serveur de diffusion groupée, ont proposé une solution, et ont mis en question les premiers mécanismes pour se joindre/quitter le bloc. John Shirron (Fore Systems) a fourni d'utiles améliorations à mes procédures originales de revalidation.

Susan Symington et Bryan Gleeson (Adaptec) ont défendu chacun de leur côté la nécessité du service fourni par les messages MARS_GROUPLIST_REQUEST/REPLY. Le nouveau schéma d'encapsulation est sorti des discussions du groupe de travail, capturées par Bryan Gleeson dans un travail en cours intérimaire (dont Keith McCloghrie (Cisco), Andy Malis (Ascom Nexion), et Andrew Smith (Bay Networks) ont été les principaux contributeurs). James Watt (Newbridge) et Joel Halpern (Newbridge) ont motivé le développement d'un format de message de contrôle de MARS plus multiprotocoles, l'éloignant de ses racines ATMARP d'origine. Ils ont aussi motivé le développement des encapsulations des chemins de données de type n° 1 et de type n° 2. Rajesh Talpade (Georgia Tech) a aidé à clarifier le besoin de la fonction MARS_MIGRATE.

Maryann Maher (ISI) a fourni de précieuses vérifications de cohérence et de mise en œuvre durant les dernières étapes du développement du document. Finalement, Jim Rubas (IBM) a fourni le pseudo code de MARS de l'Appendice F ainsi que la contre lecture des dernières étapes du développement du document.

Adresse de l'auteur

Grenville Armitage
Bellcore, 445 South Street
Morristown, NJ, 07960
USA
mél : gja@thumper.bellcore.com
téléphone : +1 201 829 2635

Références

[1] S. Deering, "Extensions d'hôte pour diffusion groupée sur IP", RFC1112, STD 5, août 1989.

[2] Juha Heinanen, "Encapsulation multiprotocole sur couche 5 d'adaptation ATM", RFC1483, juillet 1993. *(remplacée par la RFC 2684)*

[3] M. Laubach, "IP classique et ARP sur ATM", RFC1577, janvier 1994.

[4] ATM Forum, "ATM User Network Interface (UNI) Specification Version 3.1", ISBN 0-13-393828-X, Prentice Hall, Englewood Cliffs, NJ, juin 1995.

[5] D. Waitzman et autres, "Protocole d'acheminement en diffusion groupée par vecteur de distance", RFC1075, novembre 1988.

[6] M. Perez et autres, "Prise en charge de la signalisation ATM pour IP sur ATM", RFC1755, février 1995. *(P.S.)*

[7] M. Borden, E. Crawley, B. Davie et S. Batsell, "Intégration de services en temps réel dans une architecture de réseau IP-ATM", RFC1821, août 1995.

[8] ATM Forum, "ATM User-Network Interface Specification Version 3.0", Englewood Cliffs, NJ: Prentice Hall, septembre 1993.

Appendice A. Algorithmes de perçage de trous

Les mises en œuvre ont toute liberté pour se conformer au corps du présent mémoire de toutes les façons qui leurs paraissent adaptées. Le présent appendice n'est fourni qu'à titre de précision.

Une mise en œuvre de MARS peut préconstruire un ensemble de paires de <min,max> (P) qui reflètent l'espace de classe D entier, à l'exclusion de toutes les adresses actuellement prises en charge par les serveurs de diffusion groupée. Le champ <min> de la première paire DOIT être 224.0.0.0, et le champ <max> de la dernière paire doit être 239.255.255.255. La première et la dernière paires peuvent être les mêmes. Cet ensemble est mis à jour chaque fois qu'un serveur de diffusion groupée s'enregistre ou se désenregistre.

Lorsque le MARS doit effectuer un "percement de trou", il peut considérer l'algorithme suivant :

Supposons que le MARS_JOIN/LEAVE reçu par le MARS du membre de la grappe spécifiait le bloc <Emin, Emax>.

Supposons que Pmin(N) et Pmax(N) sont les champs <min> et <max> de la $N^{ème}$ paire dans l'ensemble P actuel du MARS.

Supposons que l'ensemble P a K paires. Pmin(1) DOIT être égal à 224.0.0.0, et Pmax(M) DOIT être égal à 239.255.255.255. (Si K == 1, aucun percement de trou n'est requis).

Exécutons le pseudo-code :

Créer une copie de l'ensemble P, appelons le ensemble C.

```
index1 = 1;
while (Pmax(index1) ≤ Emin)
  index1++;

index2 = K;
while (Pmin(index2) ≥ Emax)
  index2--;

si (index1 > index2)
  Exit, car l'ensemble à trous est nul.

si (Pmin(index1) < Emin)
  Cmin(index1) = Emin;

si (Pmax(index2) > Emax)
  Cmax(index2) = Emax;
```

L'ensemble C est l'ensemble "à trous" requis de blocs d'adresses.

L'ensemble C résultant conserve tous les "trous pré construits du MARS qui couvrent les serveurs de diffusion groupée, mais ont été élagués pour couvrir la section de l'espace de classe D spécifié par les valeurs <Emin,Emax> de l'hôte générateur.

L'hôte d'extréùité devrait tenir un tableau, H, des VC ouverts en ordre croiisant des adresses de classe D.

Supposons que H(x).addr est l'adresse de classe associée au VC.x.
Supposons que H(x).addr < H(x+1).addr.

Le pseudo code pour mettre à jour les VC sur la base d'un JOIN/LEAVE entrant pourrait être :

```
x = 1;
N = 1;
```

```
lorsque (x < no. de VC ouverts)
{
    lorsque (H(x).adr > max(N))
    {
        N++;
        si (N > no. de paires dans JOIN/LEAVE)
            retourne(0);
    }

    si ((H(x).adr ≤ max(N) &&
            ((H(x).addr ≥ min(N))
                perform_VC_update();
    x++;
}
```

Appendice B Minimiser l'impact de IGMP dans les environnements IPv4

La mise en œuvre d'aucune partie de cet appendice n'est exigée pour la conformité au présent document. Il n'est fourni que pour expliquer les questions qui ont été identifiées.

L'intention du paragraphe 5.1 est que les membres de la grappe n'aient que des VC en point à multipoint sortants lorsque ils envoient des données à un groupe de diffusion groupée particulier. Cependant, dans la plupart des environnements IPv4, les routeurs de diffusion groupée rattachés à une grappe vont produire périodiquement des interrogations IGMP pour vérifier si un groupe particulier a des membres. La spécification IGMP actuelle tente d'éviter que chaque membre d'un groupe réponde en insistant pour que chaque membre du groupe attende pendant une période aléatoire, et ne réponde que si aucun autre membre n'a répondu avant lui. La réponse IGMP est envoyée à l'adresse de diffusion groupée du groupe interrogé.

Malheureusement, tel qu'il est, l'algorithme IGMP sera une véritable nuisance pour les membres de la grappe qui sont essentiellement des receveurs passifs au sein d'un groupe de diffusion groupée. Ce sera déjà bien si un membre passif, sans VC sortant déjà établi avec le groupe, décide d'envoyer une réponse IGMP – causant l'établissement d'un VC alors qu'il n'est pas nécessaire de le faire. Ce n'est pas un problème fatal pour les petites grappes, mais va avoir un impact sérieux sur la capacité d'une grappe à s'y adapter.

La solution la plus évidente pour les routeurs est d'utiliser les messages MARS_GROUPLIST_REQUEST et MARS_GROUPLIST_REPLY, comme décrit au paragraphe 8.5. Cela va supprimer les interrogations IGMP régulières, d'où résultera que les membres de la grappe n'enverront qu'un rapport IGMP lors qu'ils rejoignent un groupe.

D'autres solutions existent. L'une d'elles serait de modifier l'algorithme de réponse IGMP, par exemple :

Si le membre du groupe a un VC ouvert avec le groupe, procéder selon la RFC1112 [1] (prendre un délai aléatoire entre 0 et 10 secondes pour la réponse).

Si le membre du groupe n'a pas de VC déjà ouvert avec le groupe, prendre à la place un délai aléatoire entre 10 et 20 secondes, puis procéder selon la RFC1112.

Si même un seul membre du groupe envoie au groupe au moment où est produite l'interrogation IGMP, tous les receveurs passifs vont trouver que la réponse IGMP a été transmise avant que leur délai n'expire, de sorte qu'aucun nouveau VC n'est requis. Si tous les membres du groupe sont passifs au moment de l'interrogation IGMP, une réponse va finalement arriver, mais 10 secondes plus tard que dans les circonstances conventionnelles.

La solution précédente exige de réécrire le code IGMP existant, et implique la capacité de l'entité IGMP à vérifier le statut des VC sur l'interface ATM sous-jacente. Cela ne sera vraisemblablement pas disponible à court terme.

Une solution à court terme est de fournir quelque chose comme la fonctionnalité précédente avec une "bourrique" au niveau du pilote IP/ATM parmi les membres de la grappe. S'arranger pour que le pilote IP/ATM regarde dans les paquets IP à la recherche de trafic IGMP. Si un paquet IGMP est accepté à la transmission, le pilote IP/ATM peut le mettre en mémoire tampon localement si il n'y a pas de VC déjà actif vers ce groupe. On lance un temporisateur de 10 secondes, et si une réponse IGMP est reçue pour ce groupe d'ailleurs dans la grappe, le temporisateur est remis à zéro. Si le temporisateur arrive à expiration, le pilote IP/ATM établit alors un VC vers le groupe comme il l'aurait fait pour un paquet normal en diffusion groupée IP.

Certaines mises en œuvre de réseau peuvent trouver avantageux de configurer un serveur de diffusion groupée à prendre en charge le groupe 224.0.0.1, plutôt que de s'appuyer sur un maillage. Étant donné que les routeurs de diffusion groupée IP envoient regulièrement des interrogations IGMP à cette adresse, un maillage signifira que chaque routeur va en permanence consommer un contexte AAL au sein de chaque membre de grappe. Dans les grappes desservies par plusieurs routeurs, la charge en VC au sein des commutateurs dans le réseau ATM sous-jacent va poser un problème d'adaptation.

Finalement, si un serveur de diffusion groupée est utilisé pour prendre en charge l'adresse 224.0.0.1, une autre "bourique" de niveau pilote ATM devient une solution possible au trafic de réponses IGMP. Le pilote ATM peut choisir de collecter tous les paquets IGMP sortants et de les envoyer dehors sur le VC établi pour envoyer au 224.0.0.1, sans considération de l'adresse de classe D à laquelle le message IGMP était réellement destiné. Comme tous les hôtes et routeurs doivent être membres de 224.0.0.1, les receveurs prévus vont quand même recevoir les réponses IGMP. L'impact négatif est que tous les membres de la grappe vont recevoir les réponses IGMP.

Appendice C Commentaires sur la notion de "grappe"

Le concept de grappe a été introduit à la section 1 pour deux raisons. Le terme mieux connu de "sous-réseau logique IP" est à la fois très spécifique de IP, et restreint à des frontières d'acheminement d'envoi individuel. Comme l'architecture décrire dans ce document peut être réutilisée dans des environnements non IP, un terme plus neutre était nécessaire. Comme les besoins de la diffusion groupée ne sont pas toujours liés aux mêmes portées que l'envoi individuel, il n'était pas immédiatement évident que nous limiter nous-mêmes aux LIS serait bénéfique à long terme.

On doit souligner que les grappes sont des être purement administratifs. On choisit leur taille (c'est-à-dire, le nombre de points d'extrémité qui s'enregistrent auprès du même MARS) sur la base de ses besoins de diffusion groupée, et de la consommation de ressources qu'on veut y consacrer. Plus est important le nombre d'hôtes rattachés à ATM pour lesquels on veut la prise en charge de la diffusion groupée et plus on choisira d'établir de grappes individuelles (ainsi que de routeurs de diffusion groupée pour fournir les chemins de trafic inter-grappes).

Étant donné que tous les hôtes de n'importe quel LIS n'exigent pas la prise en charge de la diffusion groupée, il devient concevable qu'on puisse allouer un seul MARS pour prendre en charge les hôtes sur plusieurs LIS. En effet, on a une grappe qui couvre plusieurs LIS, et on a réalisé des "raccourcis" pour acheminer le trafic de diffusion groupée. Dans ces circonstances, l'augmentation de la taille géographique d'une grappe peut être considérée comme une bonne chose.

Cependant, des considérations ^pratiques limitent la taille des grappes. Avoir une grappe qui s'étend sur plusieurs LIS peut n'être pas toujours une situation particulièrement "gagnante". Comme le nombre d'hôtes capables de diffusion groupée dans les LIS augmente, il devient de plus en plus vraisemblable que vous voudrez restreindre la taille d'une grappe et forcer les groupes de trafic de diffusion groupée à s'agréger à des routeurs de diffusion groupées épars à travers votre nuage ATM.

Finalement, des grappes multi-LIS exigent un certain niveau d'attention quand on déploie les routeurs de diffusion groupée IP. Dans le modèle IP classique, on a besoin de routeurs de diffusion groupée sur les bordures des LIS. Dans l'architecture de MARS, on a seulement besoin de routeurs de diffusion groupée sur les bords des grappes. Si la grappe s'étend sur pluseurs LIS, les routeurs de diffusion groupée vont alors se percevoir comme ayant une seule interface qui est simultanément rattachée à plusieurs sous-réseaux en envoi individuel. Savoir si cette situation peut fonctionner dépend des protocoles d'acheminement de diffusion groupée inter-domaines utilisés, et de la capacité des routeurs de diffusion groupée à comprendre les nouvelles relations entre les topologies d'envoi individuel et de diffusion groupée.

En l'absence de recherches plus approfondies dans ce domaine, les développeurs de réseaux conformes au présent document DOIVENT faire coincider leur grappe IP et leur LIS IP, de façon à éviter ces complications.

Appendice D Algorithme d'analyse de liste de TLV

Le pseudo-code suivant représente la façon dont pourrait être traité le format de liste de TLV décrit à la section 10 par un MARS ou client MARS.

```
liste = (mar$extoff & 0xFFFC);
si (lists == 0) exit;
lists = lists + message_base;
    alors que si (liste->Type.y != 0)
        {
            passer à (liste->Type.y)
```

```
            {
              par défaut:
                  {
                  si (liste->Type.x == 0) break;
                  si (liste->Type.x == 1) exit;
                  si (liste->Type.x == 2) log-error-and-exit;
                  }
                [...les autres traitements viennent ici..]
            }
            liste += (liste->Longueur + 4 + ((4-(liste->Longueur & 3)) % 4));
        }
retour;
```

Appendice E Résumé des valeurs de temporisateur

Cet appendice résume les divers temporisateurs ou les limites mentionnées dans le corps principal du document. Les valeurs sont spécifiées sous le format suivant : [x, y, z] indiquant une valeur minimum de x , une valeur recommandée de y, et une valeur maximum de z. Un "-" va indiquer qu'une catégorie n'a pas de valeur spécifiée. Les valeurs en minutes sont suivies par "min", les valeurs en seconds sont suivies par "s".

Temps d'inactivité du VC point à point de MARS à client MARS: [1 min, 20 min, -]

Temps d'inactivité pour les VC en multipoint venant du client. [1 min, 20 min, -]

Temps permis entre composants MARS_MULTI. [-, -, 10 s]

Gamme de temporisation de retransmission initiale aléatoire de L_MULTI_RQ/ADD : [5 s, -, 10 s]

Durée aléatoire pour établir le fanion VC_revalidate : [1 s, -, 10 s]

Intervalle de retransmission de MARS_JOIN/LEAVE : [5 s, 10 s, -]

Limite de retransmission de MARS_JOIN/LEAVE : [-, -, 5]

Durée aléatoire pour se ré-enregistrer auprès du MARS : [1 s, -, 10 s]

Attente forcée si le ré-enregistrement MARS est en boucle : [1 min, -, -]

Intervalle de transmission pour MARS_REDIRECT_MAP. [1 min, 1 min, 2 min]

Limite pour que le client manque les MARS_REDIRECT_MAP : [-, -, 4 min]

Appendice F Pseudo code pour le fonctionnement de MARS

Les mises en œuvre ont toute liberté pour se conformer au corps de ce mémoire de la façon qui leur paraît appropriée. Cet appendice ne fait qu'apporter d'éventuelles clarifications.

Une mise en œuvre de MARS pourrait être construite selon les lignes suggérées dans ce pseudo-code.

1. Principal

1.1 Initilisation

Définir une liste de serveurs comme la liste des nœuds d'extrémité sur le ServerControlVC.
Définir une liste de grappes comme la liste des nœuds d'extrémité sur le ClusterControlVC.
Définir une transposition d'hôte comme la liste des hôtes qui sont membres d'un groupe.
Définir une transposition de serveur comme la liste des hôtes (les MCS) qui desservent un groupe.
Lire configurer le fichier.
Allouer les files d'attentes de message.
Allouer les tableaux internes.

Établir la connexion passive avec le VC ouvert.
Établir le temporisateur redirect_map.
Établir les registres de journalisation.

1.2 Traitement de message

```
Forever {
 Si le message a un TLV alors {
  Si le TLV n'est pas pris en charge alors {
   traiter comme défini dans le champ Type de TLV.
  } /* TLV inconnu */
 } /* TLV présent */
 Placer le message entrant dans la file d'attente.
 Pour (tous les messages dans la file d'attente) {
  Si le message n'est pas un JOIN/LEAVE/MSERV/UNSERV avec
   mar$flags.register == 1 alors {
   Si la source du message est (pas un membre de la liste des serveurs) &&
    (pas un membre de la liste des grappes) alors {
    Abandonner le message en silence.
   }
  }
  Si (mar$pro.type n'est pas pris en charge) ou
    (l'adresse ATM de source manque) alors {
    Continuer.

  }
  Déterminer le type de message.
  Si une ERR_L_RELEASE arrive sur le ClusterControlVC alors {
   Retirer l'adresse ATM des points d’extrémité pour tous les groupes auxquels elle s'est jointe.
   Libérer le CMI.
   Continuer.
  } /* erreur sur CCVC */
  Appeler le sous-programme spécifique du traitement de message.
  Si le temporisateur redirect_map arrive à expiration {
   Appeler le sous-programme de traitement de message MARS_REDIRECT_MAP.
  } /* rediriger la fin de temporisateur */
 } /* tous les messages dans la file d'attente */
} /* fin de boucle forever */
```

2. Mécanisme de traitement de message

2.1 Messages:

- MARS_REQUEST

```
Indique pas de prise en charge du TLV par MARS_MULTI.
Si le TLV pris en charge n'est pas NULL alors {
 Indiquer la prise en charge du TLV par MARS_MULTI.
 Traiter comme requis.
} autrement { /* TLV NULL */
  Indique que le message est à envoyer sur VC privé.
  Si la source du message est un membre dela liste de serveurs alors {
   Si le groupe a une transposition d'hôte non nulle alors {
    Appeler MARS_MULTI avec la transposition d'hôte pour le groupe.
   } autrement { /* pas de groupe */
    Appeler le sous-programme du message MARS_NAK.
   } /* pas de groupe */
  } autrement { /* la source est sur la liste de grappe */
   Si le groupe a une transposition de serveur non nulle alors {
    Appeler MARS_MULTI avec la transposition de serveur pour le groupe.
   } autrement { /* membre de grappe mais pas de transposition de serveur */
    Si le groupe a une transposition d'hôte non nulle alors {
     Appeler MARS_MULTI avec la transposition d'hôte pour le groupe.
```

```
            } autrement { /* pas de groupe */
              Appeler le sous-programme de message MARS_NAK.
            } /* pas de groupe */
           } /* membre de grappe mais pas de transposition de serveur */
         } /* la source est une liste de grappe */
       } /* TLV NULL */
      Si un message existe alors {
       Envoyer le message comme indiqué.
      }
      Retour.
```

- MARS_MULTI

```
      Construire un MARS_MULTI pour la transposition spécifiée.
      Si le paramètre indique la prise en charge de TLV alors {
       Traiter le TLV comme requis.
      }
      Retour.
```

- MARS_JOIN

```
      Si (mar$flags.copy != 0) ignorer en silence le message.
      Si plus d'une seule paire <min,max> est spécifiée alors ignorer en silence le message.
      Indique que le message est à envoyer sur VC privé.
      Si (mar$flags.register == 1) alors {
       Si le nœud est déjà membre enregistré de la grappe associée au type de protocole alors { /*enregistrement antérieur*/
         Copier le CMI existant dans le MARS_JOIN.
       } autrement { /* enregistrement nouveau */
         Ajouter le nœud au ClusterControlVC.
         Ajouter le nœud à la liste de grappe.
         mar$cmi = obtenir un CMI.
       } /* enregistrement nouveau */
      } autrement autrement { /* pas un enregistrement */
       Si le groupe est dupliqué d'un MARS_JOIN précédent alors {
         mar$msn = csn en cours.
         Indique un message à envoyer sur VC privé.
       } autrement {
         Indique qu'il n'y a pas de message à envoyer.
         Si la source du message est dans une transposition de serveur alors {
           Abandonner le message en silence.
         } autrement {
           Si la première <min,max> englobe un groupe avec une transposition de serveur alors {
             Appeler le sous-programme de traitement de JOIN/LEAVE modifié.
           } autrement {
             Si le MARS_JOIN est pour un multi groupe alors {
              Appeler le sous-programme de traitement de JOIN/LEAVE Multogroupe.
             } autrement {
               Indiquer que le message est à envoyer sur ClusterControlVC.
             } /* pas pour un multi groupe */
           } /* groupe non traité par un serveur */
         } /* source du message pas dans une transposition de serveur */
         Mettre à jour less tableaux internes.
       } /* pas  un duplicata */
      } /* pas un enregistrement */

      Si un message existe alors {
       mar$flags.copy = 1.
       Envoyer le message comme indiqué.
      }
      Retour.
```

- MARS_LEAVE

```
Si (mar$flags.copy != 0) ignorer le message en silence.
Si plus d'une seule paire <min,max> est spécifiée alors ignorer le message en silence.
Indiquer le message comme à envoyer sur ClusterControlVC.
Si (mar$flags.register == 1) alors { /* désenregistrement */
  Mettre à jour les tableaux internes pour retirer l'adresse ATM du membre de tous les groupes auxquels il s'était joint.
  Sortir le point d'extrémité de ClusterControlVC.
  Sortir le point d'extrémité de la liste des grappes.
  Libérer le CMI.
  Indiquer le message comme à envoyer sur VC privé.
} autrement { /* pas un désenregistrement */
   Si le groupe est un duplicata d'un MARS_LEAVE précédent alors {
     mar$msn = csn en cours.
     Indique que le message est à envoyer sur VC privé.
   } autrement {
      Indique aucun message à envoyer.
      Si la première <min,max> englobe un groupe avec une transposition de serveur alors {
        Appeler le sous-programme de traitement de JOIN/LEAVE modifié.
      } autrement {
         Si le MARS_LEAVE est pour un multigroupe alors {
           Appeler le sous-programme de traitement de JOIN/LEAVE de multigroupe.
         } autrement {
           Indiquer que le message est à envoyer sur ClusterControlVC.
         }
       }
   Mettre à jour les tableaux internes.
  } /* pas un duplicata */
} /* pas un désenregistrement */
Si un message existe alors {
  mar$flags.copy = 1.
  Envoyer le message comme indiqué.
}
Retour.
```

- MARS_MSERV

```
 Si (mar$flags.register == 1) alors { /* enregistrement du serveur */
   Ajouter le point d'extrémité comm nœud d'extrémité à ServerControlVC.
   Ajouter le point d'extrémité à la liste de serveur.
   Indiquer que le message est à envoyer sur VC privé.
   mar$cmi = 0.
 } autrement { /* pas enregistré*/
 Si la source n'est pas enregistrée alors {
        Abandonner et ignorer le message.
        Indiquer aucun message à envoyer.
      } autrement {  /* la source est enregistrée */
         Si le MCS est déjà membre de la transposition de serveur indiquée {
            Indiquer que le message est à envoyer sur VC privé.
            mar$flags.layer3grp = 0;
            mar$flags.copy = 1.
         } autrement { /* New MCS to add. */
            Ajouter l'adresse ATM du serveur à la transposition de serveur pour le groupe.
            Indiquer que le message est à envoyer sur ServerControlVC.
            Envoyer le message comme indiqué.
            Faire une copie du message.
            Indiquer que le message est à envoyer sur ClusterControlVC.
            Si une nouvelle transposition de serveur vient d'être créée {
                Construire MARS_MIGRATE, avec le MCS comme cible.
             } autrement {
                Changer le op code en MARS_JOIN.
                mar$flags.layer3grp = 0.
                mar$flags.copy = 1.
            } /* nouvelle transposition de serveur */
         } /* Noueau MCS à ajouter. */
```

```
        } /* la source est enregistrée */
      } /* pas d'enregistrement */

      Si un message existe alors {
        Envoyer le message comme indiqué.
      }
      Retour.
```

- MARS_UNSERV

```
    Si (mar$flags.register == 1) alors { /* désenregistrer */
      Retirer l'adresse ATM du MCS de toutes les transpositions de serveur.
      Si une transposition de serveur devient nulle alors la supprimer.
      Retirer le point d'extrémité comme extrémité de ServerControlVC.
      Retirer le point d'extrémité de la liste de serveurs.
      Indique que le message est à envoyer sur VC privé.
    } autrement { /* pas un désenregistrement */
      Si la source n'est pas un membre de la liste des serveurs alors {
       Abandonner et ignorer le message.
       Indiquer aucun message à envoyer.
      } autrement {  /* la source est enregistrée */

        Si le MCS n'est pas membre de la transposition de serveur indiquée {
          Indiquer que le message est à envoyer sur VC privé.
          mar$flags.layer3grp = 0;
          mar$flags.copy = 1.
         } autrement { /* un MCS existait, il doit être retiré. */
          Retirer l'adresse ATM du MCS de la transposition de serveur indiquée.
          Si une transposition de serveur est nulle alors la supprimer.
          Indiquer que le message est à envoyer sur ServerControlVC.
          Envoyer le message comme indiqué.
          Faire une copie du message.
          Changer le op code en MARS_LEAVE.
          Indique que le message (sa copie) est à envoyer sur ClusterControlVC.
          mar$flags.layer3grp = 0;
          mar$flags.copy = 1.
        } /* un MCS existait, il doit être retiré. */
      } /* la source est enregistree */
    } /* ce n'est pas un désenregistrement */
    Si un message existe alors {
      Envoyer le message comme indiqué.
    }
    Retour.
```

- MARS_NAK

```
    Construire la commande.
    Retour.
```

- MARS_GROUPLIST_REQUEST

```
    Si (mar$pnum != 1) alors Return.
    Appeler MARS_GROUPLIST_REPLY avec la gamme et le VC de sortie.
    Retour.
```

- MARS_GROUPLIST_REPLY

```
    Construire la command pour la gamme spécifiée.
    Indiquer que le message est à envoyer sur le VC spécifié.
    Envoyer le message comme indiqué.
    Retour.
```

- MARS_REDIRECT_MAP

Inclure la propre adresse du MARSs dans le message.
Si il y a des MARS de sauvegarde alors inclure leur adresse.
Indiquer que MARS_REDIRECT_MAP est à envoyer sur ClusterControlVC.
Renvoyer le message comme indiqué.
Retour.

3. Traitement d'envoi de message

Si (le message sort de ClusterControlVC) &&
(un nouveau csn est requis) alors {
mar$msn = obtenir un CSN
}
Si (le message sort de ServerControlVC) &&
(un nouveau ssn est requis) alors {
mar$msn = obtenir un SSN
}
Retour.

4. Générateur de nombre

4.1 Numéro de séquence de grappe

Générer le prochain numéro de séquence.
Retour.

4.2 Numéro de séquence de serveur

Générer le prochain numéro de séquence.
Retour.

4.3 CMI

Les CMI sont alloués de façon univoque par membre de grappe enregistré au sein du contexte d'un type de protocole de couche 3 particulier.
Un seul nœud peut s'enregistrer plusieurs fois si il prend en charge plusieurs protocoles de couche 3.
Les CMIs alloués pour chacun de ces enregistrement peut être ou non le même.
Générer un CMI pour ce protocole.
Retour.

5. Traitement de JOIN/LEAVE modifié

Ce sous-programme traite les JOIN/LEAVE lorsque existre une transposition de serveur.

Faire une copie du message.
Changer le type de la copie en MARS_SJOIN.
Si le message est un MARS_LEAVE alors {
Changer le type de la copie en MARS_SLEAVE.
}
mar$flags.copy = 1 (copy).
Percer des trous dans le groupe <min,max> en excluant de la gamme les groupes dont le nœud qui se joint (qui quitte) est déjà (encore) membre du fait qu'il a précédement produit une adhésion individuelle au groupe.
Indiquer que le message est à envoyer sur ServerControlVC.
Si le message (sa copie) contient une ou plusieurs paires <min,max> {
Envoyer le message (sa copie) comme indiqué.
}
mar$flags.punched = 0 dans le message d'origine.
Indique que le message est à envoyer sur un VC privé.
Envoyer le message (original) comme indiqué.
Percer des trous dans le groupe <min,max> en excluant de la gamme les groupes qui sont desservis par des MCS ou desquels le nœud qui se joint (qui quitte) est déjà (encore) membre parce qu'il avait précédemment produit une adhésion individuelle au groupe.
Indiquer aue le message (original) est à envoyer sur ClusterControlVC.

```
Si (le nombre de trous percés > 0) alors { /* trous percés */
  Dans le message original faire {
   mar$flags.punched = 1.
   liste des vieux trous percés <- nouvelle liste des trous percés.
  }
} /* trous percés */
mar$flags.copy = 1.
Envoyer le message comme indiqué.
Retour.
```

5.1 Traitement de JOIN/LEAVE multi groupes

Ce sous-programme traite les JOIN/LEAVE lorsque existe un multi groupe.

```
Si (mar$flags.layer3grp) {
 Ignorer ce réglage, le remettre à zéro
}
mar$flags.copy = 1.
Faire une copie du message.
À partir de la copie percer des trous dans le groupe <min,max> en excluant de la gamme les groupes auquel ce nœud
s'est déjà joint (ou qu'il a déjà quitté).
Si (nombre de trous percés > 0) alors {
 mar$flags.punch = 0 dans le message original.
 Indique que le message original est à envoyer sur VC privé.
 Envoyer le message original comme indiqué.
 mar$flags.punch = 1 dans la copie du message.
 vielle gamme du groupe <- nouvelle liste à trous.
 Indiquer que le message est à envoyer sur ClusterControlVC.
 Envoyer la copie duof message comm indiqué.
} autrement {
  Indiquer que le message est à envoyer sur ClusterControlVC.
  Envoyer le message original comme indiqué.
} /* pas de trous percés */
Retour.
```