Groupe de travail Réseau
Request for Comments : 5044
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

P. Culley, Hewlett-Packard Company
U. Elzur, Broadcom Corporation
R. Recio, IBM Corporation
S. Bailey, Sandburst Corporation
J. Carrier, Cray Inc.
octobre 2007

Tramage aligné sur la PDU de marqueur pour la spécification de TCP

Statut du présent mémoire
Le présent document spécifie un protocole de l'Internet sur la voie de la 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 "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.
(La présente traduction incorpore l'errata 1427)

Résumé
Le tramage aligné sur la PDU de marqueur (MPA, *Marker PDU Aligned Framing*) est conçu pour fonctionner comme une "couche d'adaptation" entre TCP et le protocole de placement direct de données (DDP, *Direct Data Placement*) comme décrit dans la [RFC5041]. Il préserve la livraison fiable en ordre de TCP, tout en ajoutant la préservation des limites d'enregistrement de protocole de niveau supérieur qu'exige DDP. MPA est pleinement conforme aux RFC TCP applicables et peut être utilisé avec les mises en œuvre existantes de TCP. MPA prend aussi en charge les mises en œuvre intégrées qui combinent TCP, MPA et DDP pour réduire les exigences de mise en mémoire tampon dans la mise en œuvre et améliore les performances au niveau du système.

Table des Matières

1. Introduction

Cette Section discute des raisons de la création de MPA sur TCP et donne une vue générale du protocole.

1.1 Motivation

Le protocole de placement direct des données [RFC5041], quand il est utilisé avec TCP [RFC793], exige un mécanisme pour détecter les limites d'enregistrement. Les enregistrements DDP sont appelés des unités de données de protocole de couche supérieure (ULPDU, *Upper Layer Protocol Data Unit)* par le présent document. La capacité à localiser la limite de l'unité de données de protocole de couche supérieure est utile pour un adaptateur de réseau matériel qui utilise DDP pour placer directement les données dans la mémoire tampon d'application sur la base des informations de contrôle portées dans l'en-tête d'ULPDU. Cela peut être fait sans exiger que les paquets arrivent dans l'ordre. Les avantages potentiels de cette capacité sont d'éviter les frais d'une copie en mémoire et une plus petite exigence de mémoire pour traiter les paquets en désordre ou éliminés.

De nombreuses approches ont été proposées pour un mécanisme généralisé de tramage. Certains sont de nature probabiliste et d'autres sont déterministes. Un exemple d'approche probabiliste est caractérisé par une valeur détectable incorporée dans le flux d'octets, sans une méthode pour empêcher cette valeur ailleurs dans les données d'utilisateur. Elle est probabiliste parce que dans certaines conditions le receveur peut interpréter incorrectement les données d'application comme étant la valeur détectable. Dans ces conditions, le protocole peut échouer avec une fréquence inacceptable. Une approche déterministe est caractérisée par des contrôles incorporés à des localisations connues dans le flux d'octets. Parce que le receveur peut garantir qu'il va seulement examiner le flux des données aux localisations qui sont connues pour contenir le contrôle incorporé, le protocole ne peut jamais mal interpréter les données d'application comme étant des données de contrôle incorporées. Pour un traitement non ambigu d'un paquet décalé, une approche déterministe est préférée.

Le protocole MPA fournit un mécanisme de tramage pour DDP sur TCP qui utilise l'approche déterministe. Il permet que la localisation de la ULPDU soit déterminée dans le flux TCP même si les segments TCP arrivent en désordre.

1.2 Vue d'ensemble du protocole

La mise en couche des PDU avec MPA est montrée à la Figure 1.

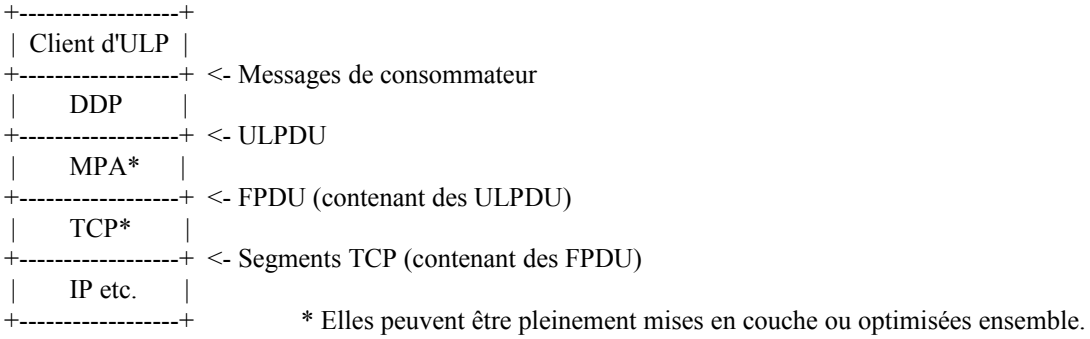

Figure 1 : Mise en couche d'ULP MPA TCP

MPA est décrit comme une couche supplémentaire au dessus de TCP et en-dessous de DDP. La séquence de fonctionnement est :

1. Une connexion TCP est établie par l'action de l'ULP. C'est fait en utilisant des méthodes non décrites par la présente spécification. L'ULP peut échanger une certaine quantité de données en mode de flux directs avant de démarrer MPA, mais n'est pas obligé de le faire.

2. Le consommateur négocie l'utilisation de DDP et MPA aux deux extrémités d'une connexion. Les mécanismes pour ce faire ne sont pas décrits dans cette spécification. La négociation peut être faite en mode de flux directs, ou par quelque autre mécanisme (comme un numéro d'accès pré-arrangé).

3. L'ULP active MPA sur chaque extrémité dans la phase de démarrage, soit comme un initiateur, soit comme un répondeur, comme déterminé par l'ULP. Ce mode vérifie l'usage de MPA, spécifie l'utilisation de CRC et de marqueurs, et permet à l'ULP de communiquer des données supplémentaires via un échange de données privées. Voir au paragraphe 7.1, "Établissement de connexion", des détails sur le processus de démarrage.

4. À la fin de la phase de démarrage, l'ULP met MPA (et DDP) en plein fonctionnement et commence à envoyer des données DDP comme décrit plus loin. Dans le présent document, les tronçons de données DDP sont appelés des ULPDU. Pour la description des données de DDP, voir la [RFC5041].

Voici la description du transfert de données quand MPA est en plein fonctionnement :

1. DDP détermine la taille maximum d'ULPDU (MULPDU) en interrogeant MPA sur cette valeur. MPA déduit cette information de TCP ou de IP, quand elle est disponible, ou choisit une valeur raisonnable.

2. DDP crée des ULPDU de la taille de MULPDU ou moins, et les passe à MPA chez l'envoyeur.

3. MPA crée une unité de données de protocole tramée (FPDU, *Framed Protocol Data Unit*) en ajoutant un en-tête, en insérant facultativement des marqueurs, et en ajoutant un champ CRC après la ULPDU et le bourrage (si il en est). MPA livre la FPDU à TCP.

4. L'envoyeur TCP met les FPDU dans le flux TCP. Si l'envoyeur est optimisé pour MPA/TCP, il segmente le flux TCP d'une façon telle qu'une limite de segment TCP soit aussi la limite d'une FPDU. TCP passe alors chaque segment à la couche IP pour transmission.

5. Le receveur peut ou non être optimisé. Si il est optimisé pour MPA/TCP, il peut séparer le passage de charge utile TCP à MPA du passage des informations d'ordre de charge utile TCP pour MPA. Dans l'un et l'autre cas, le comportement de réseau de TCP conforme aux RFC est observé chez l'envoyeur et le receveur.

6. Le receveur MPA localise et assemble les FPDU complètes dans le flux, vérifie leur intégrité, et retire les marqueurs MPA (quand ils sont présents) les champs Longueur d'ULPDU, PAD, et CRC.

7. MPA fournit alors les ULPDU complètes à DDP. MPA peut aussi séparer le passage de la charge utile MPA à DDP du passage des informations d'ordre de la charge utile MPA.

Un MPA pleinement mis en couches sur TCP est mis en œuvre comme un flux de données d'ULP pour TCP et est donc conforme aux RFC.

Un DDP/MPA/TCP optimisé utilise une couche TCP qui contient potentiellement des comportements supplémentaires comme suggéré dans ce document. Quand DDP/MPA/TCP sont optimisés à travers les couches, le comportement de TCP (en particulier la segmentation d'envoyeur) peut être différent de celui d'une mise en œuvre non optimisée, mais les différences sont dans les limites permises par les spécifications de TCP, et elle va interopérer avec un TCP non optimisé. Les comportements supplémentaires sont décrits à l'Appendice A et ne sont pas normatifs ; ils sont décrit à la couche d'interface TCP à titre indicatif. Les mises en œuvre peuvent réaliser la fonction décrite en utilisant toute méthode de leur choix, incluant des optimisations inter-couches entre TCP, MPA, et DDP.

Un envoyeur DDP/MPA/TCP optimisé est capable de segmenter les flux de données de telle façon que les segments TCP commencent par des FPDU (alignement de FPDU). Cela a des avantages significatifs pour les receveurs. Quand les segments arrivent avec des FPDU alignées, le receveur n'a généralement pas besoin de mettre en mémoire tampon de portion du segment, permettant à DDP de le placer immédiatement dans sa mémoire de destination, évitant donc les copies des mémoires tampon intermédiaires (la raison de l'existence de DDP).

Un receveur DDP/MPA/TCP optimisé permet à une mise en œuvre de DDP sur MPA de localiser le début des ULPDU qui peuvent être reçues en désordre. Il permet aussi à la mise en œuvre de déterminer si l'ULPDU entière a été reçue. Par suite, MPA peut passer des ULPDU dans le désordre à DDP pour une utilisation immédiate. Cela permet à une mise en œuvre de DDP sur MPA d'économiser une quantité significative de mémorisation intermédiaire en plaçant les ULPDU dans les bonnes localisations dans les mémoires tampon d'application quand elles arrivent, plutôt que d'attendre que l'ordre complet puisse être restauré.

La capacité d'un receveur de récupérer les ULPDU déclassées est facultative et est déclarée à l'émetteur durant le démarrage. Quand le receveur déclare qu'il ne prend pas en charge la récupération des déclassés, l'émetteur n'ajoute pas les informations de contrôle aux flux de données nécessaires pour la récupération des déclassés.

Si le receveur est pleinement mis en couches, alors MPA reçoit un flux strictement ordonné de données et ne traite pas avec des ULPDU déclassées. Dans ce cas, MPA passe chaque ULPDU à DDP quand les derniers octets arrivent de TCP, avec l'indication qu'ils sont dans l'ordre.

Les mises en œuvre de MPA qui prennent en charge la récupération des ULPDU déclassées DOIVENT prendre en charge un mécanisme pour indiquer l'ordre des ULPDU lorsque l'envoyeur les transmet et indique quand arrivent des segments intermédiaires manquants. Ces mécanismes permettent à DDP de rétablir l'ordre des enregistrements et de rapporter la livraison des messages complets (groupes d'enregistrements).

MPA vise aussi une intégrité améliorée des données. Certains utilisateurs de TCP ont noté que la somme de contrôle TCP n'est pas aussi forte qu'on pourrait le désirer (voir [CRCTCP]). Des études comme [CRCTCP] ont montré que la somme de contrôle TCP indique les segments erronnés à un taux bien supérieur à ce qu'indiqueraient les caractéristique de la liaison sous-jacente. Avec ces taux d'erreur supérieurs, les chances qu'une erreur échappe à la détection, quand on utilise seulement la somme de contrôle TCP pour l'intégrité des données, deviennent problématiques. Une vérification d'intégrité plus forte peut réduire les chances de manquer des erreurs de données.

MPA inclut une vérification de CRC pour augmenter l'intégrité des données de l'ULPDU au niveau fourni par les autres protocoles modernes, tels que SCTP [RFC4960]. Il est possible de désactiver cette vérification de CRC ; cependant, les CRC DOIVENT être activés sauf si il est clair que la connexion de bout en bout à travers le réseau a une vérification d'intégrité des données au moins aussi bonne qu'un MPA avec CRC activé (par exemple, quand IPsec est mis en œuvre de bout en bout). L'ULP de DDP attend ce niveau d'intégrité des données et donc l'ULP n'a pas à fournir sa propre intégrité des données et la récupération d'erreur dupliquées pour les données perdues.

2. Glossaire

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Consommateur : les ULP ou applications qui se tiennent au dessus de MPA et DDP. Le consommateur est responsable de l'établissement des connexions TCP, du démarrage des connexions MPA et DDP, et des opérations de contrôle générales.

CRC *(Cyclic Redundancy Check)* : contrôle de redondance cyclique.

Livraison (livré, livre) : pour MPA, la livraison est définie comme le processus d'informer DDP qu'une PDU particulière est dans l'ordre pour utilisation. Une PDU est livrée dans l'ordre exact de son envoi par l'envoyeur d'origine ; MPA utilise l'ordre du flux d'octets de TCP pour déterminer quand la livraison est possible. Ceci est spécifiquement différent de "passer la PDU à DDP", qui peut généralement avoir lieu dans n'importe quel ordre, tandis que l'ordre de livraison est strictement défini.

EMSS *(Effective Maximum Segment Size)* taille maximum de segment efficace : EMSS est la plus petite taille maximum de segment (MSS, *maximum segment size*) TCP telle que définie dans la [RFC0793], et l'unité de transmission maximum (MTU, *Maximum Transmission Unit*) [RFC1191] du chemin actuel.

FPDU *(Framed Protocol Data Unit)* unité de données de protocole tramées : unité de données créée par un envoyeur MPA.

Alignement de FPDU : propriété qu'une FPDU a son en-tête aligné avec le segment TCP, et que le segment TCP inclut un nombre entier de FPDU. Un segment TCP avec un alignement de FPDU permet un traitement immédiat des FPDU contenues sans attendre que d'autres segments TCP arrivent ou se combinent avec des segments antérieurs.

Pointeur de FPDU (FPDUPTR) : ce champ du marqueur est utilisé pour indiquer le début d'une FPDU.

Plein fonctionnement (phase de plein fonctionnement) : après l'achèvement de la phase de démarrage, MPA commence à échanger des FPDU.

Alignement d'en-tête : propriété qu'un segment TCP commence avec une FPDU. La FPDU a son en-tête aligné quand l'en-tête de FPDU est exactement au début du segment TCP (juste derrière les en-têtes TCP sur le réseau).

Initiateur : point d'extrémité d'une connexion qui envoie la trame de demande MPA, c'est-à-dire, le premier à envoyer réellement des données (qui peut n'être pas celui qui envoie le TCP SYN).

Marqueur : champ de quatre octets qui est placé dans le flux de données MPA à des intervalles d'octets fixes (tous les 512 octets).

TCP à capacité MPA : mise en œuvre de TCP qui connaît l'efficacité de FPDU MPA du receveur et est capable d'envoyer des segments TCP qui commencent avec une FPDU.

À capacité MPA : MPA est activé si le protocole MPA est visible sur le réseau. Quand l'envoyeur est à capacité MPA, il insère le tramage et les marqueurs. Quand le receveur est à capacité MPA, il interprète le tramage et les marqueurs.

Trame de demande MPA : données envoyées de l'initiateur MPA au répondeur MPA durant la phase de démarrage.

Trame de réponse MPA : données envoyées du répondeur MPA à l'initiateur MPA durant la phase de démarrage.

MPA *(Marker-based ULP PDU Aligned)* : tramage aligné sur la PDU d'ULP fondé sur le marqueur pour le protocole TCP. Le présent document définit le protocole MPA.

MULPDU : ULPDU maximum. Taille maximum actuelle de l'enregistrement qui est acceptable pour que DDP le passe à MPA pour transmission.

Nœud : appareil de calcul rattaché à une ou plusieurs liaisons d'un réseau. Un nœud dans ce contexte ne se réfère pas à une instantiation spécifique d'application ou protocole fonctionnant sur l'ordinateur. Un nœud peut consister en un ou plusieurs appareils MPA sur TCP installés dans un ordinateur hôte.

PAD *(bourrage)* : groupe de 1 à 3 octets à zéro utilisé pour compléter une FPDU à une taille exacte modulo 4.

PDU *(Protocol Data Unit)* : unité de données de protocole

Données privées : bloc de données échangées entre points d'extrémité MPA durant l'établissement initial de connexion.

Domaine de protection : concept RDMA (voir [VERBS-RDMA] et la [RFC5042]) qui lie l'utilisation de diverses ressources de point d'extrémité (accès à la mémoire, etc.) à la connexion RDMA/DDP/MPA spécifique.

RDDP *(Remote Direct Data Placement)* placement direct de données à distance : suite de protocoles incluant MPA, [RFC5040], [RFC5041], un document de sécurité globale [RFC5042], une déclaration de problèmes [RFC4297], un document d'architecture [RFC4296], et un document d'applicabilité [RFC5045].

RDMA *(Remote Direct Memory Access)* accès direct à la mémoire distante : protocole qui utilise DDP et MPA pour permettre aux applications de transférer des données directement à partir des mémoires tampon. Voir la [RFC5040].

Homologue distant : mise en œuvre du protocole MPA sur l'extrémité opposée de la connexion. Utilisé pour se référer à l'entité distante dans la description des échanges de protocole ou autres interactions entre deux nœuds.

Répondeur : point d'extrémité de connexion qui répond à une demande de connexion MPA entrante (trame de demande MAP). Ce peut n'être pas le point d'extrémité qui attendait le TCP SYN.

Phase de démarrage : échanges initiaux d'une connexion MPA qui servent à mieux identifier pleinement les points

d'extrémité MPA les uns aux autres et à passer à chaque autre point les informations d'établissement spécifiques de connexion.

ULP *(Upper Layer Protocol)* protocole de couche supérieure : couche de protocole au dessus de la couche de protocole actuellement référencée. L'ULP pour MPA est DDP [RFC5041].

ULPDU *(Upper Layer Protocol Data Unit)* unité de données de protocole de couche supérieure : enregistrement de données défini par la couche au-dessus de MPA (DDP). L'ULPDU correspond au segment DDP de DDP.

Longueur d'ULPDU : champ dans la FPDU qui décrit la longueur de l'ULPDU incluse.

3. Interactions de MPA avec DDP

DDP exige que MPA maintienne les limites d'enregistrement DDP de l'envoyeur au receveur. Quand il utilise MPA sur TCP pour envoyer des données, DDP fournit des enregistrements (ULPDU) à MPA. MPA va utiliser les capacités de transmission fiable de TCP pour transmettre les données, et va insérer les informations supplémentaires appropriées dans le flux TCP pour permettre au receveur MPA de localiser les informations de limites de l'enregistrement.

À ce titre, MPA accepte des enregistrements complets (ULPDU) provenant de DDP chez l'envoyeur et les retourne à DDP chez le receveur.

MPA DOIT encapsuler l'ULPDU de telle façon qu'il y ait exactement une ULPDU contenue dans une FPDU.

MPA sur une pile TCP standard peut généralement fournir l'alignement de FPDU avec l'en-tête TCP si la FPDU est égale à l'EMSS de TCP. Une pile MPA/TCP optimisée peut aussi maintenir l'alignement tant que la FPDU est inférieure ou égale à l'EMSS de TCP. Comme l'alignement de FPDU est généralement désiré par le receveur, DDP coopère avec MPA pour assurer que la longueur des FPDU n'excède pas la EMSS dans les conditions normales. Ceci est fait avec le mécanisme de MULPDU.

MPA DOIT fournir des informations à DDP sur la taille maximum courante de l'enregistrement acceptable à l'envoi (MULPDU). DDP DEVRAIT limiter la taille de chaque enregistrement à la MULPDU. La gamme des valeurs de MULPDU DOIT être entre 128 octets et 64 768 octets, inclus.

Le DDP envoyeur NE DOIT PAS envoyer une ULPDU de plus de 64 768 octets à MPA. DDP PEUT envoyer une ULPDU de toute taille entre un et 64768 octets ; cependant, il n'est pas EXIGÉ que MPA prenne en charge une longueur d'ULPDU supérieure à la MULPDU courante.

Bien que la longueur maximum théorique acceptée par le champ d'en-tête Longueur d'ULPDU de MPA soit de 65 535, TCP sur IP exige que la longueur maximum de datagramme IP soit de 65 535 octets. Pour permettre à MPA de prendre en charge l'alignement de FPDU, la taille maximum de la FPDU doit tenir dans un datagramme IP. Donc, la limite d'ULPDU de 64 768 octets est déduite en prenant la longueur maximum de datagramme IP, en soustrayant la longueur maximum totale de la somme de l'en-tête IPv4, de l'en-tête TCP, des options IPv4, des options TCP, et le pire cas de frais généraux MPA, et en arrondissant le résultat à une limite de 128 octets.

Noter que la MULPDU va être significativemen plus petite que le maximum théorique dans la plupart des mises en œuvre pour la plupart des circonstances, à cause de la MTU des liaisons, de l'utilisation d'en-têtes supplémentaires comme ceux requis pour IPsec, etc.

À réception, MPA DOIT passer chaque ULPDU avec sa longueur à DDP quand elle a été validée.

Si une mise en œuvre de MPA accepte de passer des ULPDU déclassées à DDP, la mise en œuvre de MPA DEVRAIT :
* Passer chaque ULPDU avec sa longueur à DDP aussitôt qu'elle a été pleinement reçue et validée.
* Fournir un mécanisme pour indiquer l'ordre des ULPDU comme l'envoyeur les a transmises. Un mécanisme possible pourrait être de fournir le numéro de séquence TCP pour chaque ULPDU.
* Fournir un mécanisme pour indiquer quand une ULPDU (et les ULPDU antérieures) est complète (livrée à DDP). Un mécanisme possible pourrait être de permettre à DDP de voir le numéro de séquence de l'accusé de réception TCP sortant courant.
* Fournir une indication à DDP que TCP a fermé ou a commencé à fermer la connexion (par exemple, reçu un FIN).

MPA DOIT fournir la version de protocole négociée avec son homologue à DDP. DDP va utiliser cette version pour régler la version dans son en-tête et pour rapporter la version à RDMA [RFC5040].

4. Phase MPA de plein fonctionnement

Les paragraphes qui suivent décrivent la signification de la phase de plein fonctionnement de MPA.

4.1 Format de FPDU

Les envoyeurs MPA créent des FPDU à partir des ULPDU. Le format d'une FPDU montré ci-dessous DOIT être utilisé pour toutes les FPDU de MPA. Pour être plus lisible, les marqueurs ne sont pas montrés à la Figure 2.

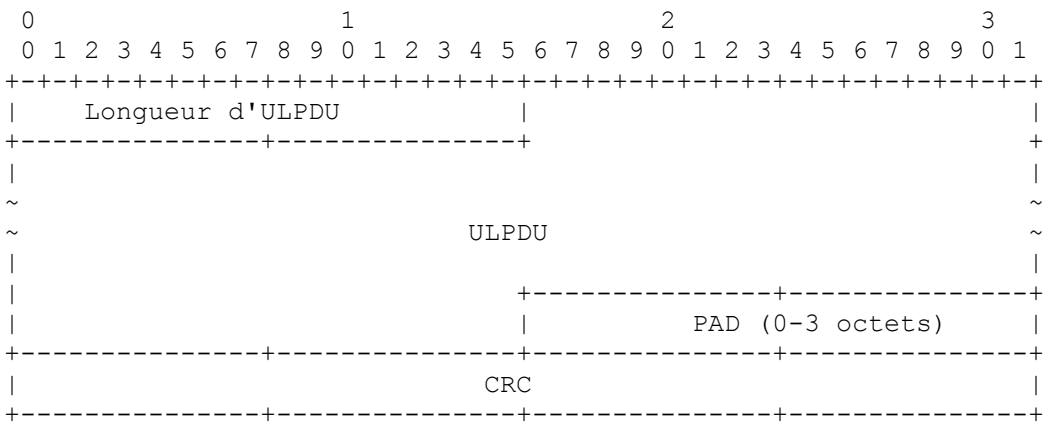

Figure 2 : Format de FPDU

Longueur d'ULPDU : 16 bits (entier non signé). C'est le nombre d'octets de l'ULPDU contenue. Il n'inclut pas la longueur de l'en-tête de FPDU lui-même, le bourrage, le CRC, ni d'aucun marqueur qui tombe dans l'ULPDU. Le champ Longueur d'ULPDU de 16 bits est assez grand pour contenir les plus grands datagrammes IP pour IPv4 ou IPv6.

PAD : le champ PAD *(bourrage)* est en queue de l'ULPDU et contient entre 0 et 3 octets de zéros. Les données de bourrage DOIVENT être réglées à zéro par l'envoyeur et ignorées par le receveur (sauf pour la vérification de CRC). La longueur du bourrage est réglée de façon à ce que la taille de la FPDU soit un multiple entier de quatre.

CRC : 32 bits. Quand les CRC sont activés, ce champ contient une valeur de vérification de CRC32c, qui est utilisée pour vérifier le contenu entier de la FPDU, en utilisant un CRC32c. Voir au paragraphe 4.4, "Calcul de CRC". Quand les CRC ne sont pas activés, ce champ est quand même présent, et peut contenir une valeur quelconque, et NE DOIT PAS être vérifié.

La FPDU ajoute un minimum de 6 octets à la longueur de l'ULPDU. De plus, la longueur totale de la FPDU va inclure la longueur de tous les marqueurs et de 0 à 3 octets de bourrage ajoutés pour arrondir la taille de ULPDU.

4.2 Format de marqueur

Le format d'un marqueur DOIT être comme spécifié à la Figure 3 :

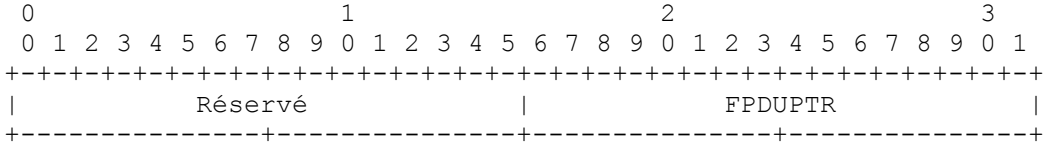

Figure 3 : Format de marqueur

Réservé : le champ Réservé DOIT être réglé à zéro à l'émission et ignoré à réception (sauf pour le calcul de CRC).

FPDUPTR : le pointeur FPDU est un pointeur relatif, long de 16 bits, interprété comme un entier non signé qui indique le nombre d'octets dans le flux TCP depuis le début du champ Longueur de ULPDU jusqu'au premier octet du marqueur entier. Les deux bits de moindre poids DOIVENT toujours être réglés à zéro à l'émission, et les receveurs DOIVENT toujours les traiter comme zéro pour les calculs (sauf pour le calcul de CRC).

4.3 Marqueurs MPA

Les marqueurs MPA sont utilisés pour identifier le début des FPDU quand les paquets sont reçus dans le désordre. Ceci est fait en situant les marqueurs à des intervalles fixes dans les flux de données (ce qui est corrélé par le numéro de séquence TCP) et en utilisant la valeur de marqueur pour localiser le début de FPDU précédent.

Tous les marqueurs MPA sont inclus dans le calcul de CRC de la FPDU conteneuse (quand CRC et marqueurs sont tous deux utilisés).

La capacité du receveur MPA à localiser les FPDU déclassées et à passer les ULPDU à DDP dépend de la mise en œuvre. MPA/DDP permet aux receveurs qui sont capables de traiter les FPDU déclassées de cette façon d'exiger l'insertion de marqueurs dans les flux de données. Quand le receveur ne peut pas traiter les FPDU déclassées de cette façon, il peut désactiver l'insertion de marqueurs chez l'envoyeur. Tous les envoyeurs MPA DOIVENT être capables de générer des marqueurs quand leur utilisation est déclarée par le receveur opposé (voir au paragraphe 7.1, "Établissement de connexion").

Quand les marqueurs sont activés, les envoyeurs MPA DOIVENT insérer un marqueur dans les flux de données à un intervalle périodique de 512 octets dans l'espace de numéros de séquence TCP. Le marqueur contient un entier non signé de 16 bits appelé FPDUPTR (pointeur FPDU).

Si la valeur du FPDUPTR n'est pas zéro, le pointeur FPDU est un pointeur arrière relatif de 16 bits. Le FPDUPTR DOIT contenir le nombre d'octets dans le flux TCP depuis le début du champ Longueur d'ULPDU jusqu'au premier octet du marqueur, sauf si le marqueur tombe entre des FPDU. Donc, la localisation du premier octet de l'en-tête de la FPDU précédente peut être déterminé en soustrayant la valeur du marqueur donné du numéro de séquence du flux d'octets courant (c'est-à-dire, le numéro de séquence TCP) du premier octet du marqueur. Noter que ce calcul DOIT tenir compte de ce que le numéro de séquence TCP pourrait être revenu à zéro entre le marqueur et l'en-tête.

Une valeur de FPDUPTR de 0x0000 est un cas particulier -- elle est utilisée quand le marqueur tombe exactement entre des FPDU (entre le champ de CRC de la FPDU précédante et le champ Longueur d'ULPDU de la prochaine FPDU). Dans ce cas, le marqueur est considéré être contenu dans la FPDU suivante ; le marqueur DOIT être inclus dans le calcul de CRC de la FPDU suivant le marqueur (si les CRC sont générés ou vérifiés). Donc, une valeur de FPDUPTR de 0x0000 signifie qu'à la suite immédiate du marqueur se trouve un en-tête de FPDU (le champ Longueur de ULPDU).

Comme toutes les FPDU sont des multiples entiers de 4 octets, les deux bits de bout du FPDUPTR calculé par l'envoyeur sont à zéro. MPA réserve ces bits, qui DOIVENT être traités comme étant à zéro pour le calcul chez le receveur.

Quand les marqueurs sont activés (voir au paragraphe 7.1, "Établissement de connexion") les marqueurs MPA DOIVENT être insérés immédiatement devant la première FPDU de phase de plein fonctionnement, et ensuite tous les 512 ième octets du flux d'octets TCP. Par suite, le premier marqueur a une valeur de FPDUPTR de 0x0000. Si le premier marqueur commence au numéro de séquence d'octet SeqStart, alors les marqueurs sont insérés de façon que le premier octet du marqueur soit au numéro de séquence d'octet SeqNum si le reste de (SeqNum - SeqStart) mod 512 est zéro. Noter que SeqNum peut revenir à zéro.

Par exemple, si le numéro de séquence TCP était utilisé pour calculer le point d'insertion du marqueur, le numéro de séquence TCP de début a peu de chances d'être zéro, et les multiples de 512 octets ont peu de chances de tomber sur un modulo 512 de zéro. Si la connexion MPA commence au numéro de séquence TCP 11, alors le 1er marqueur va commencer à 11, et les marqueurs suivants vont commencer à 523, 1035, etc.

Si une FPDU est assez grande pour contenir plusieurs marqueurs, ils DOIVENT tous pointer sur le même point dans le flux TCP : le premier octet du champ Longueur de ULPDU pour la FPDU.

Si un intervalle de marqueur contient plusieurs FPDU (les FPDU sont petites) le marqueur DOIT pointer sur le début du champ Longueur d'ULPDU pour la FPDU contenant le marqueur sauf si le marqueur tombe entre des FPDU, auquel cas le marqueur DOIT être zéro.

L'exemple suivant montre une FPDU contenant un marqueur.

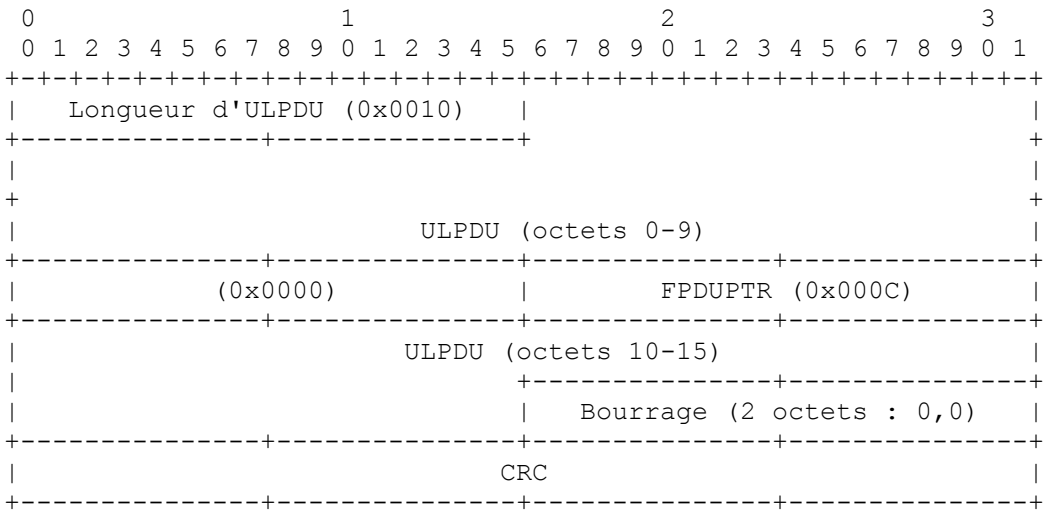

Figure 4 : Exemple de format de FPDU avec marqueur

Les receveurs MPA DOIVENT préserver les limites d'ULPDU quand ils passent les données à DDP. Les receveurs MPA DOIVENT passer les données d'ULPDU et la longueur d'ULPDU à DDP et non les marqueurs, en-têtes, et CRC.

4.4 Calcul de CRC

Une mise en œuvre de MPA DOIT prendre en charge le CRC support et DOIT soit :

1 toujours utiliser les CRC ; il N'EST PAS EXIGÉ du fournisseur MPA qu'il accepte la demande d'un administrateur de ne pas utiliser les CRC, soit

2 a) seulement indiquer une préférence pour ne pas utiliser les CRC à la demande explicite de l'administrateur du système, via une interface non définie dans la présente spécification. La configuration par défaut pour une connexion DOIT être d'utiliser les CRC.

b) désactiver la vérification de CRC (et éventuellment leur génération) si les deux points d'extrémité local et distant indiquent leur préférence pour ne pas utiliser les CRC.

Une décision administrative d'avoir la suppression du CRC de demande d'hôte NE DEVRAIT PAS être prise sauf si on a l'assurance que la connexion TCP impliquée fournit une protection contre les erreurs non détectées au moins aussi forte qu'un CRC32c de bout en bout. L'usage de bout en bout d'une vérification d'intégrité cryptographique IPsec est un des moyens de fournir une telle protection, et l'utilisation de liens de canal [RFC50506] par l'ULP peut fournir un haut niveau d'assurance que la portée de la protection IPsec est de bout en bout par rapport à l'ULP.

Le processus DOIT être invisible à l'ULP.

Après réception d'une déclaration de démarrage MPA indiquant que son homologue exige des CRC, une instance de MPA DOIT continuer de générer et vérifier les CRC jusqu'à la fin de la connexion. Si une instance de MPA a déclaré qu'elle n'exige pas les CRC, elle DOIT désactiver la vérification de CRC immédiatement après la réception d'une déclaration de mode MPA indiquant que son homologue n'exige pas non plus les CRC. Elle PEUT continuer de générer des CRC. Voir au paragraphe 7.1, "Établissement de connexion", les détails du démarrage de MPA.

Quand il envoie une FPDU, l'envoyeur DOIT inclure un champ CRC. Quand les CRC sont activés, le champ CRC dans la FPDU MPA DOIT être calculé en utilisant le polynome CRC32c de la manière décrite dans le protocole iSCSI [RFC3720] pour les résumés d'en-tête et de données.

Les champs qui DOIVENT être inclus dans le calcul de CRC lors de l'envoi d'une FPDU sont les suivants :

1) Si un marqueur ne précède pas immédiatement le champ Longueur d'ULPDU, le CRC-32c est calculé à partir du premier octet du champ Longueur d'ULPDU, sur tous l'ULPDU et marqueurs (si présents) jusqu'au dernier octet du bourrage (si présent) inclus. Si il y a un marqueur qui suit immédiatement le bourrage, le marqueur est inclus dans le

calcul de CRC pour cette FPDU.

2) Si un marqueur précède immédiatement le premier octet du champ Longueur d'ULPDU de la FPDU, (c'est-à-dire, le marqueur est tombé entre deux FPDU, et est donc obligé d'être inclus dans la seconde FPDU) le CRC-32c est calculé à partir du premier octet du marqueur, sur l'en-tête Longueur d'ULPDU, toute la ULPDU et les marqueurs (si il en est de présents) jusqu'au dernier octet de bourrage (si il en est de présents) inclus.

3) Après le calcul du CRC-32c, la valeur résultante est placée dans le champ CRC à la fin de la FPDU.

Quand une FPDU est reçue, et que la vérification de CRC est activée, le receveur DOIT d'abord effectuer ce qui suit :

1) Calculer le CRC de la FPDU entrante de la même façon que défini ci-dessus.

2) Vérifier que la valeur du CRC-32c calculée est la même que celle du CRC-32c reçue trouvée dans le champ CRC de FPDU. Sinon, le receveur DOIT traiter la FPDU comme étant invalide.

La procédure pour traiter les FPDU invalides est couverte à la Section 8, "Sémantique des erreurs".

Ce qui suit est un dépôt annoté en hexadécimal d'un exemple de FPDU envoyée comme première FPDU du flux. À ce titre, il commence par un marqueur. La FPDU contient une ULPDU de 42 octets (un exemple de segment DDP) qui à son tour contient 24 octets de l'ULPDU contenue, qui est une charge de données toutes à zéro. Le CRC32c a été correctement calculé et peut être utilisé comme référence. Voir dans les [RFC5040] et [RFC5041] la définition des champs Contrôle DDP, File d'attente, MSN, MO, et Données envoyées.

Compte d'octets	Contenu	Annotation
0000	00	marqueur : réservé
0001	00	
0002	00	marqueur : FPDUPTR
0003	00	
0004	00	Longueur d'ULPDU
0005	2a	
0006	41	Champ Contrôle DDD, envoyé avec le fanion Last établie
0007	43	
0008	00	Réservé (position de STag DDP sans STag)
0009	00	
000a	00	
000b	00	
000c	00	File d'attente DDP = 0
000d	00	
000e	00	
000f	00	
0010	00	MSN DDP = 1
0011	00	
0012	00	
0013	01	
0014	00	MO DDP = 0
0015	00	
0016	00	
0017	00	
0018	00	Données DDP envoyées (24 octets de zéros)
...		
002f	00	
0030	52	CRC32c
0031	23	
0032	99	
0033	83	

Figure 5 : dépôt annoté en hexadécimal d'une FPDU

Voici un exemple envoyé comme seconde FPDU d'un flux où la première FPDU (qui n'est pas montrée ici) avait une longueur de 492 octets et a aussi un envoi à File d'attente 0 avec le fanion Last établi. Cet exemple contient un marqueur.

Compte d'octets	Contenu	Annotation
01ec	00	Longueur
01ed	2a	
01ee	41	Champ Contrôle DDP : envoyé avec le fanion Last établi
01ef	43	
01f0	00	Réservé (position de STag DDP sans STag)
01f1	00	
01f2	00	
01f3	00	
01f4	00	File d'attente DDP = 0
01f5	00	
01f6	00	
01f7	00	
01f8	00	MSN DDP = 2
01f9	00	
01fa	00	
01fb	02	
01fc	00	MO DDP = 0
01fd	00	
01fe	00	
01ff	00	
0200	00	marqueur : réservé
0201	00	
0202	00	marqueur : FPDUPTR
0203	14	
0204	00	Données DDP envoyées (24 octets de zéros)
...		
021b	00	
021c	84	CRC32c
021d	92	
021e	58	
021f	98	

Figure 6 : dépôt annoté en hexadécimal d'une FPDU avec marqueur

4.5 Considérations sur la taille de FPDU

MPA définit l'unité maximum de données de protocole de couche supérieure (MULPDU, *Maximum Upper Layer Protocol Data Unit*) comme la taille de la plus grande ULPDU tenant dans une FPDU. Pour un segment TCP vide, la MULPDU est l'EMSS moins les frais généraux de FPDU (6 octets) moins l'espace pour les marqueurs et les octets de bourrage.

La longueur maximum d'ULPDU pour une seule ULPDU quand des marqueurs sont présents DOIT être calculée comme :

MULPDU = EMSS - (6 + 4 * plafond(EMSS / 512) + EMSS mod 4)

La formule ci-dessus tient compte du pire cas de nombre de marqueurs.

La longueur maximum d'ULPDU pour une seule ULPDU quand des marqueurs NE sont PAS présents DOIT être calculée comme :

MULPDU = EMSS - (6 + EMSS mod 4)

Comme optimisation supplémentaire de l'efficacité, une mise en œuvre de MPA PEUT ajuster dynamiquement la MULPDU (voir à la Section 5 les compromis entre latence et efficacité du réseau). Quand une ou plusieurs FPDU sont déjà mises en paquet dans un segment TCP, la MULPDU PEUT être réduite en conséquence.

DDP DEVRAIT fournir des ULPDU qui sont aussi grandes que possible, mais inférieures ou égales à la MULPDU.

Si la mise en œuvre de TCP a besoin d'ajuster l'EMSS pour prendre en charge des changements de MTU ou des

changemennrs d'options TCP, la valeur de MULPDU est changée en conséquence.

Dans certaines situations rares, l'EMSS peut se réduire en dessous de 128 octets. Si cela se produit, l'envoyeur de MPA sur TCP NE DOIT PAS réduire la MULPDU en dessous de 128 octets et n'est pas obligé de suivre les règles de segmentation du paragraphe 5.1 et de l'Appendice A.

Si une ou plusieurs FPDU sont déjà mises en paquet dans un segment TCP, de telle sorte que la place restante est inférieure à 128 octets, MPA NE DOIT PAS fournir une MULPDU inférieure à 128. Dans ce cas, MPA va normalement fournir une MULPDU pour le prochain segment de taille entière, mais peut quand même empaqueter la prochaine FPDU dans la petite place restante, pourvu que la prochaine FPDU soit assez petite pour y tenir.

La valeur 128 est choisie pour donner aux concepteurs de DDP de la place pour l'en-tête DDP et des données d'utilisateur.

5. Interactions de MPA avec TCP

Les paragraphes qui suivent décrivent les interactions de MPA avec TCP. Cette Section discute de l'utilisation d'une pile TCP standard mise en couche avec MPA rattaché au-dessus d'une prise TCP. La discussion de l'utilisation d'un TCP optimisé à capacité MPA avec une mise en œuvre MPA qui tire parti des optimisations supplémentaires est faite à l'Appendice A.

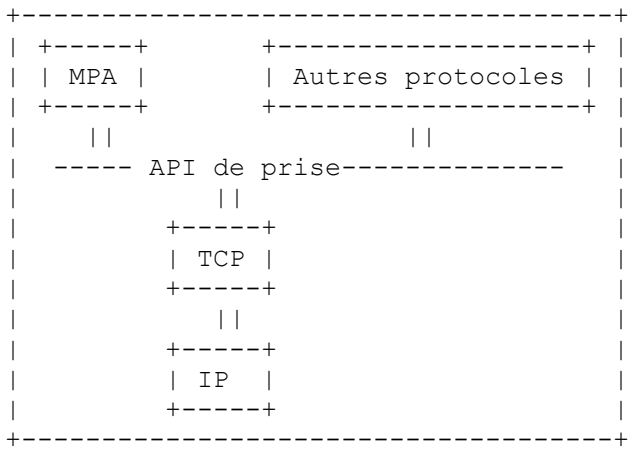

Figure 7 : mise en œuvre pleinement mise en couches

La mise en œuvre pleinement mise en couches est décrite pour être complet ; cependant, l'utilisateur est averti que la probabilité réduite d'alignement de FPDU lors de la transmission avec cette mise en œuvre va tendre à introduire des frais généraux plus importants chez les receveurs optimisés. De plus, le manque de traitement de réceptions en désordre va significativement réduire la valeur de DDP/MPA en imposant de plus forts frais généraux de mise en mémoire tampon et de copie chez le receveur local.

5.1. Transmetteurs MPA avec TCP en couches standard

Les transmetteurs MPA DEVRAIENT calculer une MULPDU comme décrit au paragraphe 4.5. Si la mise en œuvre de TCP permet que l'EMSS soit déterminée par MPA, cette valeur devrait être utilisée. Si le côté transmetteur de mise en œuvre de TCP n'est pas capable de rapporter l'EMSS, MPA DEVRAIT utiliser la valeur courante de MTU pour établir une taille de FPDU probable, en tenant compte des diverses tailles d'en-tête attendues.

Les transmetteurs MPA DEVRAIENT aussi utiliser toutes les facilités que présente la pile TCP pour faire que le transmetteur TCP commence les segments TCP aux limites de FPDU. Plusieurs FPDU PEUVENT être mises en paquet dans un seul segment TCP comme déterminé par le calcul d'EMSS pour autant qu'elles soient entièrement contenues dans le segment TCP.

Par exemple, passer des mémoires tampon de FPDU dimensionnées à l'EMSS courante à la prise TCP et utiliser l'option de prise TCP_NODELAY pour désactiver l'algorithme de Nagle [RFC896] va généralement résulter en ce que beaucoup des segments commencent avec une FPDU.

Il est reconnu que divers effets peuvent causer la perte d'un alignement de FPDU. Voici quelques uns de ces effets :

* Les ULPDU sont plus petites que la MULPDU. Si elles sont envoyées dans un flux continu, l'alignement de FPDU va être perdu. Noter qu'une utilisation attentive d'une MULPDU dynamique peut aider dans ce cas ; la MULPDU pour les futures FPDU peut être ajustée pour rétablir l'alignement avec les segments fondés sur l'EMSS courante.

* L'envoi de suffisamment de données pour que la fenêtre de réception de TCP atteigne sa limite. TCP peut envoyer un plus petit segment pour remplir exactement la fenêtre de réception.

* L'envoi de données quand TCP fonctionne contre la fenêtre d'encombrement. Si TCP ne colle pas à la fenêtre d'encombrement dans les segments, il peut transmettre un plus petit segment pour remplir exactement la fenêtre de réception.

* Changer l'EMSS à cause de diverses options de TCP, ou changements de la MTU.

Si l'alignement de FPDU avec les segments TCP est perdu pour une raison quelconque, l'alignement est récupéré après une coupure dans la transmission où les mémoires tampon d'envoi de TCP sont vidées. De nombreux modèles d'usage de TCP pour DDP/MPA vont inclure de telles coupures.

Il est EXIGÉ des receveurs MPA qu'ils soient capable de fonctionner correctement même si l'alignement est perdu (voir la Section 6).

5.2. Receveurs MPA avec TCP en couches standard

Les receveurs MPA vont obtenir les données de TCP dans le flux ordonné usuel. Les receveurs DOIVENT identifier les limites de FPDU en utilisant le champ Longueur d'ULPDU, comme décrit à la Section 6. Les receveurs PEUVENT utiliser des marqueurs pour vérifier la cohérence des limites de FPDU, mais ils NE sont PAS obligés d'examiner les marqueurs pour déterminer les limites de FPDU.

6. Identification de FPDU de receveur MPA

Un receveur MPA DOIT d'abord vérifier la FPDU avant de passer l'ULPDU à DDP. Pour ce faire, le receveur DOIT :
* localiser sans ambiguïté le début de la FPDU,
* vérifier son CRC (si la vérification de CRC est activée).

Si les conditions ci-dessus sont vérifiées, le receveur MPA passe l'ULPDU à DDP.

Pour détecter sans ambiguïté le début de la FPDU, une des méthodes suivantes DOIT être utilisée :

1 : Dans un flux TCP ordonné, le champ Longueur d'ULPDU dans la FPDU courante, quand la FPDU a un CRC valide, peut être utilisé pour identifier le début de la prochaine FPDU.

2 : Pour les receveurs MPA/TCP optimisés qui prennent en charge la réception de FPDU déclassées (voir au paragraphe 4.3, "Marqueurs MPA") un marqueur peut toujours être utilisé pour localiser le début d'une FPDU (dans les FPDU avec un CRC valide). Comme la localisation du marqueur est connue dans le flux (espace de numéros de séquence) le marqueur peut toujours être trouvé.

3 : Ayant trouvé une FPDU au moyen d'un marqueur, un receveur MPA/TCP optimisé peut trouver les FPDU contiguës suivantes en utilisant les champs Longueur d'ULPDU (à partir des FPDU avec des CRC valides) pour établr la limite de FPDU suivante.

Le champ Longueur d'ULPDU (voir la Section 4) DOIT être utilisé pour déterminer si la FPDU entière est présente avant de transmettre la ULPDU à DDP.

Le calcul de CRC est discuté au paragraphe 4.4.

7. Sémantique de connexion

7.1 Établissement de connexion

MPA exige que le consommateur active MPA, et toutes les améliorations de TCP pour MPA, sur une demie connexion TCP à la même localisation dans le flux d'octets chez l'envoyeur et chez le receveur. Ceci est exigé afin que le schéma de marqueur localise correctement les marqueurs (si ils sont activés) et localise correctement la première FPDU.

MPA, et toutes les améliorations de TCP pour MPA sont activés par l'ULP dans les deux directions en une fois à un point d'extrémité.

Ceci peut être accompli de plusieurs manières, et est laissé à la décision de l'ULP de DDP :

* L'ULP de DDP PEUT exiger le démarrage de DDP sur MPA immédiatement après l'établissement de la connexion TCP. Ceci a l'avantage qu'aucune négociation de mode de flux n'est nécessaire. Un exemple d'un tel protocole est montré à la Figure 10 : Exemple de négociation immédiate de démarrage.

  Ceci peut être accompli en utilisant un accès bien connu, ou un protocole de localisation de service pour localiser un accès approprié sur lequel DDP sur MPA est supposé fonctionner.

* L'ULP de DDP PEUT négocier le début de DDP sur MPA quelque temps après un démarrage TCP normal, en utilisant les échanges de données de flux TCP sur la même connexion. L'échange établit que DDP sur MPA (ainsi que d'autres ULP) va être utilisé, et localise exactement le point dans le flux d'octets où MPA va commencer à fonctionner. Noter qu'un tel protocole de négociation sort du domaine d'application de la présente spécification. Un exemple simplifié d'un tel protocole est montré à la Figure 9 : Exemple de négociation de démarrage retardé.

Un point d'extrémité MPA opère en deux phases distinctes.

La phase de démarrage est utilisée pour vérifier l'établissement correct de MPA, du CRC d'échange et de la configuration de marqueur, et facultativement pour passer des données privées entre les points d'extrémité avant de réaliser une connexion DDP. Durant cette phase, des trames spécifiquement formatées sont échangées comme des flux d'octets TCP sans utiliser de CRC ou de marqueurs. Durant cette phase, un point d'extrémité DDP n'a pas besoin d'être "lié" à la connexion MPA. En fait, le choix d'un point d'extrémité DDP et de ses paramètres de fonctionnement peut n'être pas connu jusqu'à ce que le consommateur ait examiné les données privées fournies (si il en est).

La second phase distincte est le plein fonctionnement durant lequel les FPDU sont envoyées en utilisant toutes les règles pertinentes (CRC, marqueurs, restrictions de MULPDU, etc.). Un point d'extrémité DDP DOIT être "lié" à la connexion MPA à l'entrée de cette phase.

Quand des données privées sont passées entre les ULP dans la phase de démarrage, l'ULP est chargé d'interpréter ces données, et ensuite de placer MPA en plein fonctionnement.

Note : Le texte suivant différencie les deux points d'extrémité en initiateur et répondeur à l'appel. Ceci est assez arbitraire et NE se rapporte PAS au démarrage de TCP (séquence SYN, SYN/ACK). L'initiateur est le côté qui envoie d'abord dans la séquence de démarrage MPA (la trame de demande MPA).

Note : La possibilité qu'il soit permis aux deux points d'extrémité de faire une connexion au même moment, parfois appelée une connexion active/active, a été considérée par le groupe de travail et rejetée. Il y a plusieurs motifs à cette décision. L'une est que les applications qui ont besoin de cette facilité sont peu nombreuses (aucune autre que théorique au moment de la rédaction de ce document). Une autre est que la facilité créait des difficultés de mise en œuvre, en particulier avec les concepts de "double pile" décrits plus loin. Un dernier problème se rapporte au rejet de connexions au démarrage qui aurait exigé au moins un type de trame supplémentaire, et plus d'actions de récupération, compliquant le protocole. Bien qu'aucun de ces problèmes ne soit insurmontable, le groupe et les développeurs n'étaient pas motivés pour travailler à résoudre ces questions. Le protocole inclut une méthode pour détecter ces tentatives de démarrage actif/actif afin qu'elles puissent être rejetées et qu'une erreur soit rapportée.

L'ULP est responsable de la détermination de quel côté est initiateur ou répondeur. Pour les ULP de type client/serveur, c'est facile. Pour les ULP d'homologue à homologue (qui pourraient utiliser un démarrage TCP de style actif/actif) un mécanisme (non défini dans la présente spécification) doit être établi, ou des données en mode flux direct échangées avant le démarrage de MPA pour déterminer quel côté commence comme initiateur et qui commence en mode répondeur MPA.

7.1.1 Format de trame de demande et réponse MPA

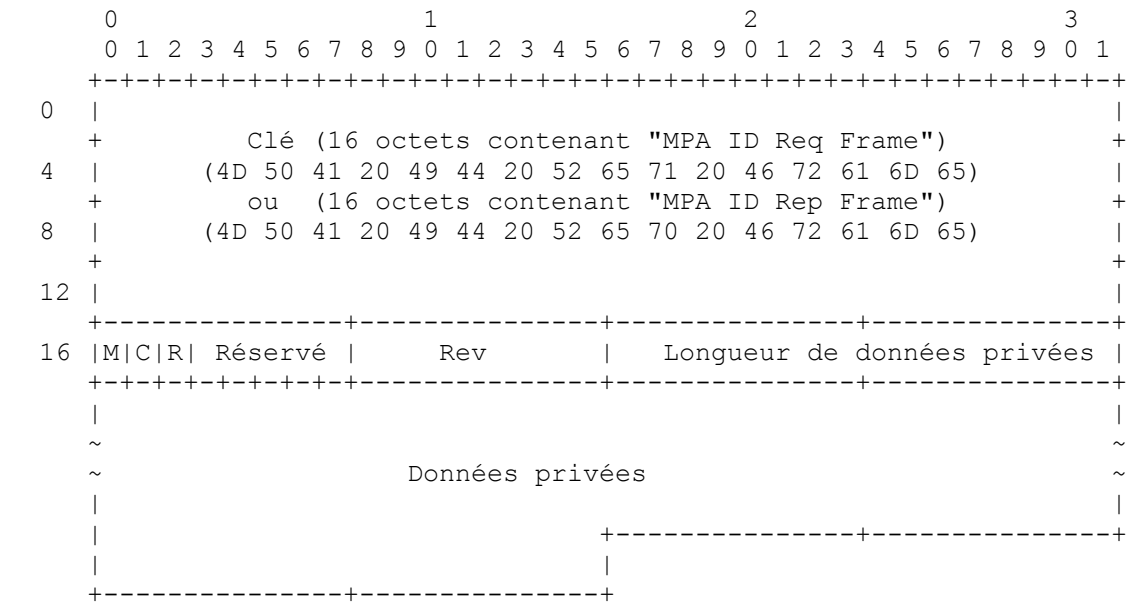

Figure 8 : Trame MPA de demande/réponse

Clé : ce champ contient la "clé" utilisée pour valider que l'envoyeur est un envoyeur MPA. Les envoyeurs en mode initiateur DOIVENT régler ce champ à la valeur fixe de "MPA ID Req Frame" ou (dans l'ordre des octets) 4D 50 41 20 49 44 20 52 65 71 20 46 72 61 6D 65 (en hexadécimal). Les receveurs en mode répondeur DOIVENT vérifier que cela a la même valeur, et clore la connexion et rapporter une erreur en local si toute autre valeur est détectée. Les envoyeurs en mode répondeur DOIVENT régler ce champ à la valeur fixe de "MPA ID Rep Frame" ou (dans l'ordre des octets) 4D 50 41 20 49 44 20 52 65 70 20 46 72 61 6D 65 (en hexadécimal). Les receveurs en mode initiateur DOIVENT vérifier que ce champ a la même valeur, clore la connexion et rapporter une erreur en local si toute autre valeur est détectée.

M : ce bit déclare l'usage EXIGÉ de marqueur d'un point d'extrémité. Quand ce bit est "1" dans une trame de demande MPA, l'initiateur déclare que les marqueurs sont EXIGÉS dans les FPDU envoyées du répondeur. Quand il est réglé à "1" dans une trame de réponse MPA, ce bit déclare que les marqueurs sont EXIGÉS dans les FPDU envoyées de l'initiateur. Quand dans une trame de demande MPA ou trame de réponse MPA reçue la valeur est "0", des marqueurs NE DOIVENT PAS être ajoutés au flux de données par ce point d'extrémité. Quand il est réglé à "1", des marqueurs DOIVENT être ajoutés comme décrit au paragraphe 4.3, "Marqueurs MPA".

C : ce bit déclare l'usage de CRC préféré d'un point d'extrémité. Quand ce champ est "0" dans la trame de demande MPA et la trame de réponse MPA, les CRC NE DOIVENT PAS être vérifiés et n'ont pas besoin d'être générés par l'un ou l'autre point d'extrémité. Quand ce bit est "1" dans la trame de demande MPA ou la trame de réponse MPA, les CRC DOIVENT être générés et vérifiés par les deux points d'extrémité. Noter que même quand il n'est pas utilisé, le champ CRC reste présent dans la FPDU. Quand les CRC ne sont pas utilisés, le champ CRC DOIT être considéré comme valide pour la vérification de FPDU sans considération de son contenu.

R : ce bit est réglé à zéro, et non vérifié à réception dans la trame MPA. Dans la trame de réponse MPA, ce bit est le bit de connexion rejetée, réglé par les ULP répondeurs pour indiquer l'acceptation "0", ou le rejet "1", des paramètres de la connexion fournis dans les données privées.

Réservé : ce champ est réservé pour une utilisation future. Il DOIT être réglé à zéro à l'envoi, et non vérifié à réception.

Rev : ce champ contient la révision de MPA. Pour cette version de la spécification, les envoyeurs DOIVENT régler ce champ à un. Les receveurs MPA conformes à la présente version de la spécification DOIVENT vérifier ce champ. Si le receveur MPA ne peut pas interopérer avec la version reçue, il DOIT alors clore la connexion et rapporter une erreur en local. Autrement, le receveur MPA devrait rapporter la version reçue à l'ULP.

Longueur de données privées : ce champ DOIT contenir la longueur en octets du champ Données privées. Une valeur de zéro indique qu'il n'y a pas du tout de champ Données privées présent. Si le receveur détecte que le champ Longueur de données privées ne correspond pas à la longueur du champ Données privées, ou si la longueur du champ Données

privées excède 512 octets, le receveur DOIT clore la connexion et rapporter une erreur en local. Autrement, le receveur MPA devrait passer la valeur de Longueur de données privées et les données privées à l'ULP.

Données privées : ce champ peut contenir toute valeur définie par les ULP ou peut n'être pas présent. Le champ Données privées DOIT être entre 0 et 512 octets. Les ULP définissent comment dimensionner, régler, et valider ce champ dans ces limites. L'usage des données privées est expliqué au paragraphe 7.1.4.

7.1.2 Règles de démarrage de connexion

Les règles suivantes s'appliquent à la phase de démarrage de connexion MPA :

1. Quand MPA est démarré en mode initiateur, la mise en œuvre de MPA DOIT envoyer une trame de demande MPA valide. La trame de demande MPA PEUT inclure des données privées fournies par l'ULP.

2. Quand MPA est démarré en mode répondeur, la mise en œuvre de MPA DOIT attendre qu'une trame de demande MPA soit reçue et validée avant d'entrer en plein fonctionnement MPA/DDP.

   Si la trame de demande MPA est formatée de façon impropre, la mise en œuvre DOIT clore la connexion TCP et sortir de MPA.

   Si la trame de demande MPA est formatée de façon appropriée mais que les données privées ne sont pas acceptables, la mise en œuvre DEVRAIT retourner une trame de réponse MPA avec le bit Connexion rejetée réglé à "1" ; la trame de réponse MPA PEUT inclure des données privées fournies par l'ULP ; la mise en œuvre DOIT sortir de MPA, laissant la connexion TCP ouverte. L'ULP peut clore TCP ou utiliser la connexion pour un autre objet.

   Si la trame de demande MPA est formatée de façon appropriée et si les données privées sont acceptables, la mise en œuvre DEVRAIT retourner une trame de réponse MPA avec le bit Connexion rejetée réglé à "0" ; la trame de réponse MPA PEUT inclure des données privées fournies par l'ULP ; et le répondeur DEVRAIT se préparer à interpréter toutes les données reçues comme des FPDU et passer toutes les ULPDU reçues à DDP.

Note : comme la capacité du receveur à traiter les marqueurs est inconnue jusqu'à ce que les trames de demande et réponse aient été reçues, l'envoi de FPDU avant que cela se produise n'est pas possible.

Note : l'exigence d'attendre une trame de demande avant d'envoyer une trame de réponse est un choix de conception. Elle impose une séquence bien ordonnée d'événements à chaque extrémité, et évite d'avoir à spécifier comment traiter les situations où les deux extrémités démarrent en même temps.

3. Les mises en œuvre en mode initiateur MPA DOIVENT recevoir et valider une trame de réponse MPA.

   Si la trame de réponse MPA est formatée de façon impropre, la mise en œuvre DOIT clore la connexion TCP et sortir de MPA.

   Si la trame de réponse MPA est formatée de façon appropriée mais si les données privées ne sont pas acceptables, ou si le bit Connexion rejetée est réglé à "1", la mise en œuvre DOIT sortir de MPA, laissant la connexion TCP ouverte. L'ULP peut clore TCP ou utiliser la connexion pour un autre objet.

   Si la trame de réponse MPA est formatée de façon appropriée et si les données privées sont acceptables, et si le bit Connexion rejetée est réglé à "0", la mise en œuvre DEVRAIT entrer dans la phase de plein fonctionnement MPA/DDP, interprétant toutes les données reçues comme des FPDU et en envoyant les ULPDU DDP comme des FPDU.

4. Les mises en œuvre en mode répondeur MPA DOIVENT recevoir et valider au moins une FPDU avant d'envoyer des FPDU ou marqueurs.

Note : cette exigence est présente pour donner à l'initiateur le temps que son receveur passe en plein fonctionnement avant qu'une FPDU arrive, évitant de potentielles conditions de concurrence chez l'initiateur. Cela a aussi fait l'objet de débats dans le groupe de travail avant qu'un consensus soit atteint. Éliminer cette exigence permettrait un démarrage plus rapide dans certains types d'applications. Cependant, cela rendrait aussi certaines mises en œuvre (en particulier de "double pile") plus difficiles.

5. Si une "clé" reçue ne correspond pas à la valeur attendue (voir au paragraphe 7.1.1, "Format de trame de demande et réponse MPA") la connexion TCP/DDP DOIT être close, et une erreur retournée à l'ULP.

6. Les champs Données privées reçus peuvent être utilisés par les consommateurs à l'une et l'autre extrémité pour valider la connexion et établir DDP ou d'autres paramètres d'ULP. L'ULP initiateur PEUT clore la connexion TCP/MPA/DDP par suite de la validation des champs Données privées. Le répondeur DEVRAIT retourner une trame de réponse MPA avec le bit "Connexion rejetée " réglé à "1" si la validation des données privées n'est pas acceptable à l'ULP.

7. Quand la première FPDU est à envoyer, si les marqueurs sont activés, les premiers octets envoyés sont le marqueur spécial 0x00000000, suivi par le début de la FPDU (le champ Longueur d'ULPDU de la FPDU). Si les marqueurs ne sont pas activés, les premiers octets envoyés sont le début de la FPDU (le champ Longueur d'ULPDU de la FPDU).

8. Les mises en œuvre de MPA DOIVENT utiliser la différence entre la trame de demande MPA et la trame de réponse MPA pour vérifier des démarrages incorrects "initiateur/initiateur". Les mises en œuvre DEVRAIT mettre un temporisateur en attente pour la trame de demande MPA quand elle est démarrée en mode répondeur, pour détecter les démarrages incorrects "répondeur/répondeur".

9. Les mises en œuvre de MPA DOIVENT valider le champ Longueur de données privées. La mémoire tampon qui reçoit le champ Données privées DOIT être assez grande pour recevoir les données ; la quantité de données privées NE DOIT PAS excéder la longueur de données privées ou la mémoire tampon d'application. Si une des conditions ci-dessus échoue, la trame de démarrage DOIT être considérée comme improprement formatée.

10. Les mises en œuvre de MPA DEVRAIENT utiliser une temporisation raisonnable pour attendre l'ensemble complet de trames de démarrage ; cela empêche certaines attaques de déni de service. Les ULP DEVRAIENT mettre en œuvre une temporisation raisonnable pour attendre les FPDU, ULPDU, et messages de niveau application pour se garder contre les défaillances d'application et certaines attaques de déni de service.

7.1.3 Exemple de démarrage de séquence retardé

Diverses séquences de démarrage sont possibles quand on utilise MPA sur TCP. Voici un exemple d'un démarrage MPA/DDP qui se produit après que TCP a fonctionné pendant un certain temps et a échangé une certaine quantité de flux de données. Cet exemple n'utilise pas de données privées (un exemple qui le fait est montré au paragraphe 7.1.4.2, "Exemple de démarrage immédiat utilisant des données privées") bien qu'il soit parfaitement légal d'inclure des données privées. Noter que comme l'exemple n'utilise pas de données privées, il n'y a pas d'interactions d'ULP entre la réception des "trames de démarrage" et la mise de MPA en plein fonctionnement.

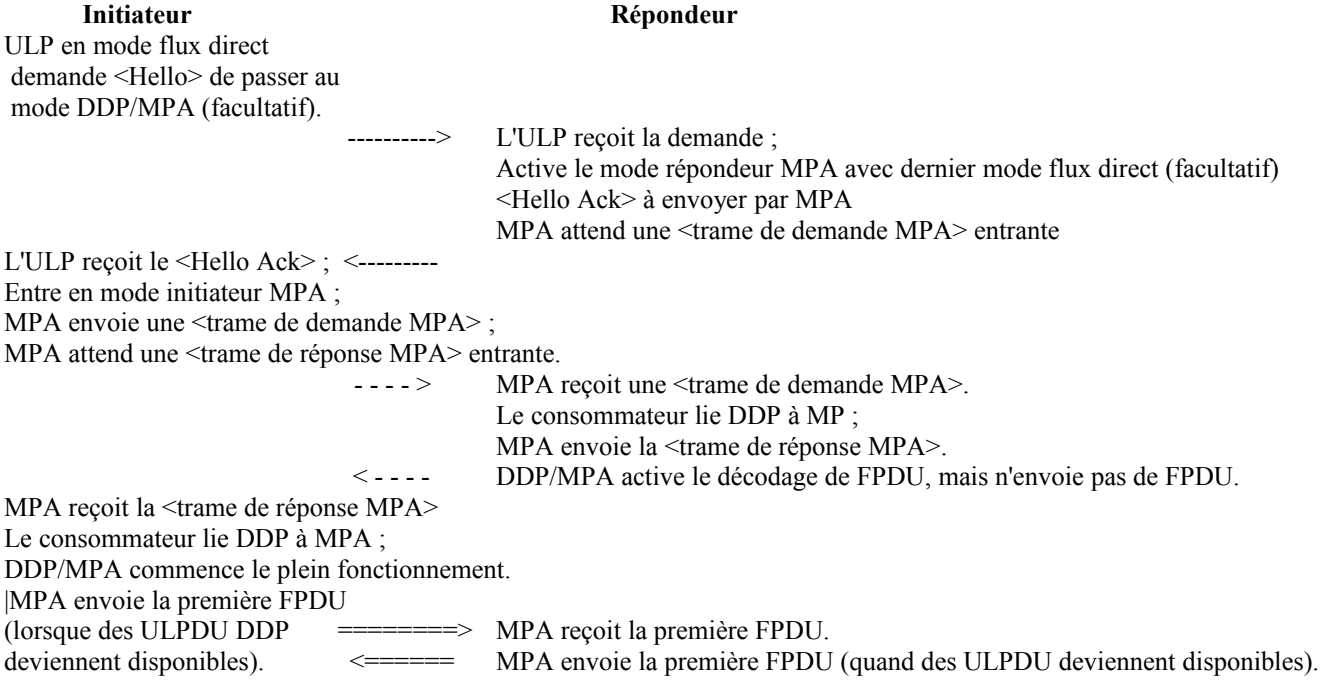

Figure 9 : Exemple de négociation de démarrage retardé

On décrit ci-après un exemple de séquence de démarrage retardé :

* Le côté actif et le côté passif démarrent une connexion TCP de la façon usuelle, probablement en utilisant des API de prises. Ils échangent une certaine quantité de données en mode flux directs. À un certain moment, un côté (l'initiateur MPA) envoie des données en mode flux direct qui disent effectivement "Hello, passons en mode MPA/DDP".

* Quand le côté distant (le répondeur MPA) obtient ce message de mode de flux direct, le consommateur va envoyer un dernier message en mode de flux direct qui dit effectivement "J'accuse réception de ton Hello, et suis maintenant en mode répondeur MPA". L'échange de ces messages établit le point exact dans le flux TCP où MPA est activé. Le consommateur répondeur active MPA dans le mode répondeur et attend le message initial de démarrage MPA.

* Le consommateur répondeur va activer le démarrage MPA dans le mode initiateur dans lequel il envoie la trame de demande MPA. On suppose qu'aucun message de données privées n'est nécessaire pour cet exemple, bien qu'il soit possible de le faire. Le MPA initiateur (et consommateur) va aussi attendre que la connexion MPA soit acceptée.

* Le MPA répondeur va recevoir la trame de demande MPA initiale et va informer le consommateur que ce message est arrivé. Le consommateur peut alors accepter la connexion MPA/DDP ou clore la connexion TCP.

* Pour accepter la demande de connexion, le consommateur répondeur va utiliser une API appropriée pour lier les connexions TCP/MPA à un point d'extrémité DDP, activant donc le plein fonctionnement de MPA/DDP. Dans le processus de passage en plein fonctionnement, MPA envoie la trame de réponse MPA. MPA/DDP attend la première FPDU entrante avant d'envoyer des FPDU.

* Si les données TCP initiales n'étaient pas une trame de demande MPA formatée de façon appropriée, MPA va immédiatement clore ou réinitialiser la connexion TCP.

* Le MPA initiateur va recevoir la trame de réponse MPA et va rapporter ce message au consommateur. Le consommateur peut alors accepter la connexion MPA/DDP, ou clore ou réinitialiser la connexion TCP pour interrompre le processus.

* Pour déterminer que la connexion est acceptable, le consommateur initiateur va utiliser une API appropriée pour lier les connexions TCP/MPA à un point d'extrémité DDP, mettant donc MPA/DDP en plein fonctionnement. MPA/DDP va commencer à envoyer des messages DDP comme des FPDU MPA.

7.1.4 Utilisation de données privées

Ce paragraphe est indicatif par nature, en ce qu'il suggère une méthode pour qu'un ULP puisse traiter les échanges d'information précédant la connexion DDP.

7.1.4.1 Motivation

Les protocoles RDMA antérieurement développés fournissaient des données privées via des mécanismes hors bande. Par suite, de nombreuses applications attendent maintenant qu'une forme de données privées soit disponible à l'usage de l'application avant d'établir la connexion DDP/RDMA. Voici des exemples d'utilisation de données privées.

Un point d'extrémité RDMA (appelé une paire de files d'attente (QP, *Queue Pair*) dans InfiniBand et [VERBS-RDMA]) doit être associé à un domaine de protection. Aucune opération de réception ne peut être envoyée au point d'extrémité avant qu'il soit associé à un domaine de protection. Bien sûr, aussi bien dans InfiniBand et dans les verbes proposés de RDMA/DDP [VERBS-RDMA] un point d'extrémité/QP est créé dans un domaine de protection.

Dans certaines applications, le choix du domaine de protection dépend de l'identité du client d'ULP distant. Par exemple, si une session d'utilisateur exige plusieurs connexions, il est très souhaitable que toutes ces connexions utilisent un seul domaine de protection. Note : l'utilisation des domaines de protection est discutée plus en détails dans la [RFC5042].

InfiniBand, les API DAT [DAT-API], et [IT-API] fournissent toutes à l'ULP côté actif la possibilité de présenter des données privées quand il demande une connexion. Ces données sont passées à l'ULP pour lui permettre de déterminer si il accepte la connexion, et si il en est ainsi, avec quel point d'extrémité (et implicitement quel domaine de protection).

Les données privées peuvent aussi être utilisées pour s'assurer que les deux extrémités de la connexion ont configuré leurs

points d'extrémité RDMA de façon compatible sur la questton de la capacité RDMA Read (voir la [RFC5040]). D'autres utilisations spécifiques de l'ULP sont aussi présumées, comme d'établir l'identité du client.

Les données privées sont aussi permises pour quand on accepte la connexion, permettre l'achèvement de toute négociation sur les resources RDMA et pour d'autres raisons au niveau de l'ULP.

Il y a plusieurs façons possibles d'échanger ces données privées. Par exemple, la spécification InfiniBand inclut un protocole de gestion de connexion qui permet qu'une petite quantité de données privées soit échangée en utilisant des datagrammes avant de commencer réellement la connexion RDMA.

Le présent document permet que de petites quantités de données privées soient échangées au titre de la séquence de démarrage de MPA. Les champs réels de données privées sont portés dans la trame de demande MPA et la trame de réponse MPA.

Si de plus grandes quantités de données privées ou plus de négociations sont nécessaires, des messages TCP en mode de flux directs peuvent être échangés avant d'activer MPA.

7.1.4.2 Exemple de démarrage immédiat utilisant des données privées

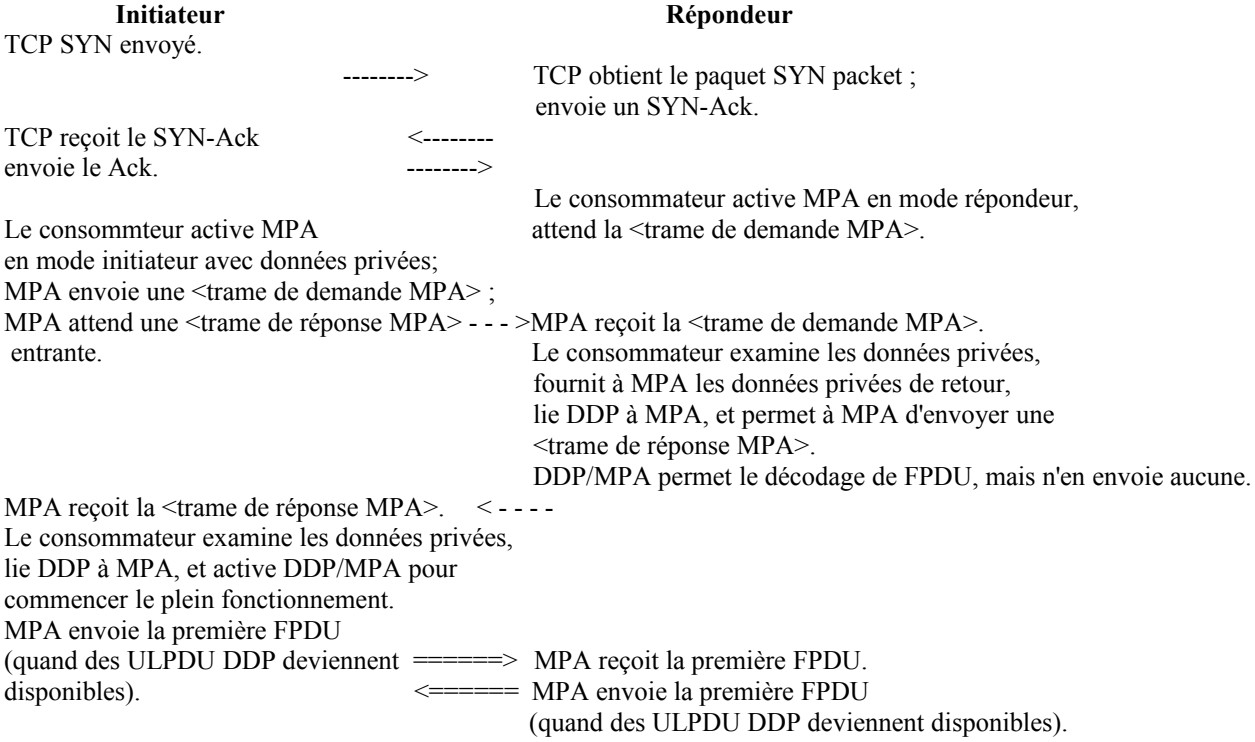

Figure 10 : Exemple de négociation de démarrage immédiat

Note : l'ordre exact de quand MPA est démarré dans la séquence de connexion TCP dépend de la mise en œuvre ; le diagramme ci-dessus montre une séquence possible. Aussi, le "Ack" de l'initiateur au "SYN-Ack" du répondeur peut être combiné dans le même segment TCP qui contient la trame de demande MPA (comme c'est permis par TCP).

L'exemple de séquence de démarrage immédiat est décrit ci-dessous :

* Le côté passif (consommateur répondeur) va écouter sur l'accès de destination TCP, pour indiquer qu'il est prêt à accepter une connexion.

* Le côté actif (consommateur initiateur) va demander une connexion à un point d'extrémité TCP (qui s'attendait à se mettre à niveau à MPA/DDP/RDMA et attendait des données privées) à une adresse et accès de destination.

* Le consommateur initiateur va initier une connexion TCP avec l'accès de destination. L'acceptation/rejet de la connexion va se faire selon les règles normales d'établissement de connexion TCP.

* Le côté passif (consommateur répondeur) va recevoir la demande de connexion TCP comme d'habitude en permettant que les portiers TCP normaux, comme des serveurs INETD et TCP, exercent leurs fonctions normales de sauvegarde/enregistrement. À l'acceptation de TCP, le consommateur répondeur va activer MPA en mode répondeur et attendre le message initial de démarrage de MPA.

* Le consommateur initiateur va activer le démarrage de MPA en mode initiateur pour envoyer une trame de demande MPA avec inclus son message de données privées à envoyer. Le MPA initiateur (et consommateur) va aussi attendre que la connexion MPA soit acceptée, et toutes les données privées retournées.

* Le MPA répondeur va recevoir la trame initiale de demande MPA avec le message de données privées et va passer les données privées au consommateur. Le consommateur peut alors accepter la connexion MPA/DDP, clore la connexion TCP, ou rejeter la connexion MPA avec un message de retour.

* Pour accepter la demande de connexion, le consommateur répondeur va utiliser une API appropriée pour lier les connexions TCP/MPA à un point d'extrémité DDP, activant donc MPA/DDP en plein fonctionnement. Dans le processus de passage en plein fonctionnement, MPA envoie la trame de réponse MPA, qui inclut les données privées fournies par le consommateur, contenant toute réponse appropriée du consommateur. MPA/DDP attend la première FPDU entrante avant d'envoyer des FPDU.

* Si les données TCP initiales n'étaient pas une trame de demande MPA formatée de façon appropriée, MPA va immédiatement clore ou réinitialiser la connexion TCP.

* Pour rejeter la demande de connexion MPA, le consommateur répondeur va envoyer une trame de réponse MPA avec toutes données privées fournies par l'ULP (avec la raison du rejet) avec le bit "Connexion rejetée" réglé à "1", et peut clore la connexion TCP.

* Le MPA initiatiateur va recevoir la trame de réponse MPA avec le message de données privées et va rapporter ce message au consommateur, incluant les données privées fournies. Si le bit "Connexion rejetée" est réglé à "1", MPA va clore la connexion TCP et sortir. Si le bit "Connexion rejetée" est réglé à "0", et en déterminant d'après les données privées de la trame de réponse MPA que la connexion est acceptable, le consommateur initiateur va utiliser une API appropriée pour lier les connections TCP/MPA à un point d'extrémité DDP, mettant donc MPA/DDP en plein fonctionnement. MPA/DDP va commencer à envoyer des messages DDP comme des FPDU MPA.

7.1.5 Mises en œuvre de "double pile"

Les mises en œuvre de MPA/DDP sont généralement supposées faire partie d'une architecture de "double pile". Une des piles est le TCP traditionnel, avec une interface de programmation d'application (API, *Application Programming Interface*) de prises. La seconde pile est celle de MPA/DDP avec sa propre API, et potentiellement un logiciel ou matériel distinct pour traiter les données de MPA/DDP. Bien sûr, les mises en œuvre peuvent varier, de sorte que les commentaires suivants sont seulement de nature indicative.

L'utilisation des deux piles offre des avantages :

L'établissement de la connexion TCP est généralement fait avec la pile TCP. Cela permet d'utiliser les mécanismes usuels de nommage et d'adressage. Cela signifie aussi que tout mécanisme utilisé pour "durcir" l'établissement de connexion contre les menaces pour la sécurité est aussi utilisé au démarrage de MPA/DDP.

Certaines applications peuvent avoir été conçues à l'origine pour TCP, mais sont "améliorées" pour utiliser MPA/DDP après qu'une négociation révèle la capacité de le faire. Le processus de négociation a lieu en mode de flux direct de TCP, en utilisant les API TCP usuelles.

Certaines nouvelles applications, conçues pour RDMA ou DDP, ont quand même besoin d'échanger des données avant de démarrer MPA/DDP. Cet échange peut être de longueur ou complexité arbitraire, mais consiste souvent en seulement une petite quantité de données privées, peut-être un seul message. Utiliser TCP en mode de flux direct pour cet échange permet que cela soit fait en utilisant des méthodes bien comprises.

Le principal inconvénient de l'utilisation de deux piles est la conversion d'une connexion TCP active entre elles. Ce processus doit être fait avec prudence pour empêcher des pertes de données.

Pour éviter certains des problèmes de l'utilisation d'une architecture de "double pile", les restrictions supplémentaires suivantes peuvent être requises par la mise en œuvre :

1. L'activation de la pile DDP/MPA DEVRAIT être faite seulement quand aucun flux de données entrant n'est attendu. Ceci est normalement géré MPA/DDP par l'ULP. Quand il suit la séquence de démarrage recommandée, le côté répondeur entre en mode DDP/MPA, envoie les dernières données en mode flux direct, et ensuite attend la trame de demande MPA. Aucune donnée supplémentaire en mode flux direct n'est attendue. L'ULP côté initiateur reçoit les dernières données en mode flux direct, et entre ensuite en mode DDP/MPA. Là encore, aucune donnée supplémentaire en mode flux direct n'est attendue.

2. DDP/MPA PEUT fournir la capacité d'envoyer le "dernier message de flux direct" au titre de sa fonction d'activation de répondeur DDP/MPA. Cela permet à la pile DDP/MPA de gérer plus facilement la conversion au mode DDP/MPA (et éviter des problèmes avec un retour très rapide de la trame de demande MPA du côté initiateur).

Note : sans considération de l'architecture de "pile" utilisée, les règles de TCP DOIVENT être suivies. Par exemple, si des données du réseau sont perdues, resegmentées, ou réarrangées, TCP DOIT récupérer de façon appropriée même quand cela se produit lors de l'échange des piles.

7.2 Suppression normale de connexion

Chaque demie connexion de MPA se termine quand DDP clôt la demie connexion TCP correspondante.

Un mécanisme DEVRAIT être fourni par MPA à DDP pour que DDP soit averti qu'une clôture en douceur de la connexion TCP a été reçue par TCP (par exemple, FIN a été reçu).

8. Sémantique des erreurs

Les erreurs suivantes DOIVENT être détectées par MPA et les codes DEVRAIENT être fournis à DDP ou autre consommateur :

Code	Erreur
1	Connexion TCP close, terminée, ou perdue. Cela inclut la perte par fin de temporisation, trop d'essais, réception de RST, ou FIN.
2	Le CRC MPA reçu ne correspond pas à la valeur calculée pour la FPDU.
3	Dans le cas où le CRC est valide, les champs Marqueur MPA reçu (si il est activé) et Longueur d'ULPDU ne s'accordent pas sur le début d'une FPDU. Si le début de FPDU déterminé à partir des champs Longueur d'ULPDU précédents ne correspondent pas avec la position du marqueur MPA, MPA DEVRAIT livrer une erreur à DDP. Il peut n'être pas possible de faire cette vérification lorsque un segment arrive, mais la vérification DEVRAIT être faite quand un trou créant une séquence en désordre est bouché et chaque fois qu'un marqueur pointe sur une FPDU déjà identifiée. Il est FACULTATIF pour un receveur de vérifier chaque marqueur, si plusieurs marqueurs sont présents dans une FPDU, ou si le segment est reçu dans l'ordre.
4	Réception d'une trame de demande ou réponse MPA invalide. Dans ce cas, la connexion TCP DOIT être close immédiatement. DDP et les autres ULP devraient traiter cela de façon similaire au code 1.

Quand les conditions 2 ou 3 ci-dessus sont détectées, une mise en œuvre de MPA/TCP optimisée PEUT choisir d'éliminer en silence le segment TCP plutôt que de rapporter l'erreur à DDP. Dans ce cas, le TCP envoyeur va reessayer le segment, généralement en corrigeant l'erreur, sauf si le problème était à la source. Dans ce cas, la source va généralement excéder le nombre d'essais et terminer la connexion.

Une fois que MPA a livré une erreur d'un type quelconque, il NE DOIT PAS passer ou livrer de FPDU supplémentaire sur cette demie connexion.

Pour les codes d'erreur 2 et 3, MPA NE DOIT PAS clore la connexion TCP à la suite d'un rapport d'erreur. Clore la connexion est de la responsabilité de l'ULP de DDP.

Noter que comme MPA ne va pas livrer de FPDU sur une demie connexion suite à une erreur détectée sur le côté récepteur de cette connexion, l'ULP de DDP est supposé supprimer la connexion. Cela ne peut pas se produire avant qu'un ou plusieurs derniers messages soient transmis sur la demie connexion opposée. Cela permet qu'un message de diagnostic

d'erreur soit envoyé.

9. Considérations sur la sécurité

Cette Section discute des considérations de sécurité pour MPA.

9.1 Considérations sur la sécurité spécifiques du protocole

Les vulnérabilités de MPA aux attaques de tiers ne sont pas plus grandes que celles de tout autre protocole fonctionnant sur TCP. Un tiers, en envoyant des paquets dans le réseau qui sont livrés à un receveur MPA, pourrait lancer diverses attaques pour tirer parti de la façon dont fonctionne MPA. Par exemple, un tiers pourrait envoyer des paquets aléatoires qui seraient valides pour TCP, mais ne contiendraient pas d'en-tête de FPDU. Un receveur MPA rapporte une erreur à DDP quand un paquet arrive et ne peut pas être validé comme FPDU quand il est bien situé sur une limite de FPDU. Un tiers pourrait aussi envoyer des paquets qui sont valides pour TCP, MPA, et DDP, mais ne ciblent pas de mémoires tampon valides. Ces types d'attaques résultent en fin de compte en la perte de connexion et donc deviennent un type d'attaque de déni de service (DOS, *Denial Of Service*). Des mécanismes de sécurité de la communication comme IPsec [RFC2401], [RFC4301] peuvent être utilisés pour empêcher de telles attaques.

Indépendamment de la façon dont MPA fonctionne, un tiers pourrait utiliser des messages ICMP pour réduire la MTU du chemin à une taille si petite que les performances seraient probablement sévèrement impactées. La vérification de gamme sur les tailles de MTU de chemin dans les paquets ICMP peut être utilisée pour empêcher de telles attaques.

Les [RFC5040] et [RFC5041] sont utilisées pour contrôler, lire, et écrire des mémoires tampon de données sur les réseaux IP. Donc, les paquets de contrôle et de données de ces protocoles sont vulnérables aux attaques d'usurpation d'identité, d'altération et de divulgation d'informations examinées ci-dessous. De plus, la connexion de/vers un point d'extrémité non autorisé ou non authentifié est un problème potentiel avec la plupart des applications qui utilisent RDMA, DDP, et MPA.

9.1.1 Usurpation d'identité

Les attaques en usurpation d'identité peuvent être lancées par l'homologue distant ou par un attaquant dans le réseau. Une attaque en usurpation d'identité fondée sur le réseau s'applique à tous les homologues distants. Parce que le flux MPA exige un flux TCP dans l'état ÉTABLI, certains types de formes traditionnelles d'attaque sur le réseau ne s'appliquent pas -- une prise de contact de bout en bout doit s'être produite pour établir le flux MPA. Donc, la seule forme d'usurpation d'identité qui s'applique est quand un nœud distant peut à la fois envoyer et recevoir des paquets. Mais même avec cette limitation, le flux est quand même exposé aux attaques d'usurpation d'identité suivantes.

9.1.1.1 Se faire passer pour un autre

Un attaquant dans le réseau peut se faire passer pour un homologue légal MPA/DDP/RDMAP (en usurpant une adresse IP légale) et établir un flux MPA/DDP/RDMAP avec la victime. L'authentification de bout en bout (c'est-à-dire, l'authentification IPsec ou d'ULP) fournit une protection contre cette attaque.

9.1.1.2. Capture de flux

La capture de flux se produit quand un attaquant dans le réseau suit la phase d'établissement de flux, et attend que la phase d'authentification (si une telle phase existe) soit achevée avec succès. Il peut alors usurper l'adresse IP et rediriger le flux provenant de la victime sur sa propre machine. Par exemple, un attaquant peut attendre qu'une authentification iSCSI soit achevée avec succès, et capturer le flux iSCSI.

La meilleure protection contre cette forme d'attaque est la protection de l'intégrité et l'authentification de bout en bout, comme avec IPsec, pour empêcher l'usurpation d'identité. Une autre option est de fournir une sécurité physique. La discussion de la sécurité physique sort du domaine d'application de ce document.

9.1.1.3 Attaque par interposition

Si un attaquant dans le réseau a la capacité de supprimer, injecter, répéter, ou modifier les paquets qui vont quand même être acceptés par MPA (par exemple, le numéro de séquence TCP est correct, la FPDU est valide, etc.) alors le flux peut

être exposé à une attaque par interposition. L'attaquant pourrait utiliser les services de la [RFC5040] et de la [RFC5041] pour lire le contenu de la mémoire tampon de données associée, pour modifier le contenu de la mémoire tampon de données associée, ou pour désactiver l'accès à la mémoire tampon. D'autres attaques sur la séquence d'établisssement de la connexion et même sur TCP peuvent être utilisées pour causer un déni de service. La seule contre-mesure pour cette forme d'attaque est de sécuriser le flux MPA/DDP/RDMAP (c'est-à-dire, la protection de l'intégrité) ou de tenter de fournir la sécurité physique pour empêcher les attaques de type interposition.

La meilleure protection contre cette forme d'attaque est la protection de l'intégrité et l'authentification de bout en bout, comme avec IPsec, pour empêcher l'usurpation d'identité ou l'altération. Si l'authentification et la protection de l'intégrité au niveau du flux ou de la session ne sont pas utilisées, alors une attaque par interposition peut se produire, permettant l'usurpation d'identité et l'altération des données.

Une autre approche est de restreindre l'accès à seulement le sous-réseau/liaison local et de fournir un mécanisme pour limiter l'accès, comme la sécurité physque ou 802.1.x. Ce modèle est un scénario de déploiement extrêmement limité et ne sera pas examiné plus en détails ici.

9.1.2 Espionnage

Généralement parlant, la confidentialité du flux protège contre l'espionnage. L'authentification et la protection de l'intégrité du flux et/ou session sont une contre-mesure contre les diverses attaques d'usurpation et d'altération. L'efficacité de l'authentification et de l'intégrité contre une attaque spécifique dépend de si l'authentification est au niveau machine (comme celle fournie par IPsec) ou de l'ULP.

9.2 Introduction aux options de sécurité

Les services de sécurité suivants peuvent être appliqués à un flux MPA/DDP/RDMAP :
1. Confidentialité de session : protège contre l'espionnage.
2. Authentification de source des données par paquet : protège contre les attaques d'usurpation d'identité suivantes : se faire passer pour une autre personne dans le réseau, capture de flux, et interposition.
3. Intégrité par paquet : protège contte l'altération faite par la modification des FPDU dans le réseau (affectant indirectement le contenu de mémoire tampon sur les services DDP).
4. Séquençage de paquet : protège contre les attaques en répétition, qui sont un cas particulier de l'attaque d'altération.

Si un flux MPA/DDP/RDMAP peut être soumis à des attaques d'usurpation d'identité, ou des attaques de capture de flux, il est recommandé que le flux soit authentifié, protégé en intégrité, et protégé contre les attaques en répétition. Il peut utiliser la protection de la confidentialité pour protéger de l'espionnage (dans le cas où le flux MPA/DDP/RDMAP traverse un réseau public).

IPsec est capable de fournir les services de sécurité ci-dessus pour le trafic IP et TCP.

Les ULP peuvent être capables de fournir une partie des services de sécurité ci-dessus. Voir dans la [RFC5056] des informations supplémentaires sur une approche prometteuse appelée "lien de canal". D'après la [RFC5056] : "Le concept de lien de canal permet aux applications de prouver que les points d'extrémité de deux canaux sûrs à des couches de réseau différentes sont les mêmes en liant l'authentification à un canal à la protection de session à l'autre canal. L'utilisation des liens de canal permet aux applications de déléguer la protection de session aux couches inférieures, ce qui peut améliorer significativement les performances de certaines applications."

9.3 Utilisation de IPsec avec MPA

IPsec peut être utilisé pour protéger contre les attaques d'injection de paquets mentionnées ci-dessus. Comme IPsec est conçu pour sécuriser les paquets IP individuels, MPA peut fonctionner par dessus IPsec sans changement. Les paquets IPsec sont traités (par exemple, vérification d'intégrité et déchiffrés) dans l'ordre de leur réception, et un receveur MPA va traiter les FPDU déchiffrées contenues dans ces paquets de la même manière que des FPDU contenues dans des paquets IP non sécurisés.

Les mises en œuvre de MPA DOIVENT appliquer IPsec comme décrit au paragraphe 9.4. L'utilisation de IPsec relève des ULP et des administrateurs.

9.4 Exigences pour l'encapsulation IPsec de MPA/DDP

Le groupe de travail "IP Storage" a passé un temps significatif et n'a pas ménagé ses efforts pour définir les exigences normatives de IPsec pour la mémorisation IP [RFC3723]. Des portions de cette spécification sont applicables à une grande variété de protocoles, incluant la suite de protocoles RDDP. Pour ne pas dupliquer cet effort, une mise en œuvre de MPA sur TCP DOIT suivre les exigences définies au paragraphe 2.3 et à la Section 5 de la RFC 3723, incluant les références normatives associées à ces sections.

De plus, comme un matériel d'accélération IPsec peut seulement être capable de traiter un nombre limité d'associations de sécurité (SA, *Security Association*) de phase 2 actives du protocole d'échange de clés Internet (IKE, *Internet Key Exchange Protocol*) les messages de phase 2 Supprime PEUVENT être envoyés pour des SA inactives, comme moyen de garder le nombre de SA actives de phase 2 à un minimum. La réception d'un message Supprime IKE de phase 2 NE DOIT PAS être interprété comme une raison de supprimer un flux DDP/RDMA. Il est préférable de laisser le flux actif, et si du trafic supplémentaire est envoyé sur lui, de mettre une autre SA IKE de phase 2 pour le protéger. Cela évite de potentiellement activer et désactiver continuellement des flux.

Les exigences de IPsec pour RDDP se fondent sur la version de IPsec spécifiée dans la [RFC2401] et les RFC en relation, comme dans le profil de la [RFC3723], en dépit de l'existence d'une version plus récente de IPsec spécifiée dans la [RFC4301] et les RFC qui s'y rapportent. Une des applications précoces importantes des protocoles RDDP est leur utilisation avec iSCSI [RFC5046] ; les exigences de IPsec de RDDP suivent celles de IPsec afin de faciliter cet usage en permettant qu'un profil commun de IPsec soit utilisé avec iSCSI et les protocoles RDDP. À l'avenir, la RFC 3723 pourrait être mise à jour avec la version plus récente de IPsec ; les exigences de sécurité de IPsec d'une telle mise à jour devraient s'appliquer uniformément aux protocoles iSCSI et RDDP.

Noter qu'il y a de sérieux problèmes de sécurité si IPsec n'est pas mis en œuvre de bout en bout. Par exemple, si IPsec est mis en œuvre comme un tunnel au milieu du réseau, tous les hôtes entre l'homologue et l'appareil de tunnelage IPsec peuvent librement attaquer le flux non protégé.

10. Considérations relatives à l'IANA

Aucune action de l'IANA n'est requise par le présent document. Si un accès bien connu est choisi comme mécanisme pour identifier un DDP sur MPA sur TCP, l'accès bien connu doit être enregistré auprès de l'IANA. Parce que l'utilisation de l'accès est spécifique de DDP, l'enregistrement de l'accès auprès de l'IANA est laissé à DDP.

Appendice A. Mises en œuvre optimisée de TCP à capacité MPA

Cet Appendice est seulement pour information et NE fait PAS partie de la norme. Cet Appendice traite de lignes directrices de mise en œuvre de TCP à capacité MPA optimisées. Il est destiné aux mises en œuvre qui veulent envoyer/recevoir autant de trafic que possible de façon alignée et sans copie.

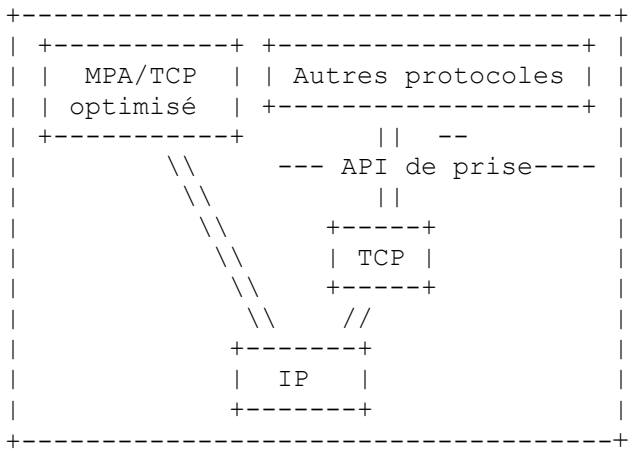

Figure 11 : Mise en œuvre optimisée de MPA/TCP

Le diagramme ci-dessus montre une mise en œuvre potentielle de blocs. Le sous-système réseau dans le diagramme peut prendre en charge des connexions traditionnelles fondées sur la prise utilisant l'API normale comme montré sur le côté droit du diagramme. Les connexions pour DDP/MPA/TCP fonctionnent en utilisant les facilités montrées sur le côté gauche du diagramme.

Les connexions DDP/MPA/TCP peut être démarrées en utilisant les facilités montrées sur le côté gauche en utilisant des API convenables, ou elles peuvent être initiées en utilisant les facilités montrées sur le côté droit et transitées au côté gauche au point de l'établissement de connexion où MPA passe à la "phase MPA/DDP de plein fonctionnement", comme décrit au paragraphe 7.1.2.

Les mises en œuvre optimisées de MPA/TCP (côté gauche du diagramme et décrites ci-dessous) sont seulement applicables à MPA. Toutes les autres applications de TCP continuent d'utiliser les piles et interfaces TCP standard montrés sur e côté droit du diagramme.

A.1 Émetteurs optimisés MPA/TCP

Les diverses RFC TCP permettent un choix considérable pour la segmentation d'un flux TCP. Afin d'optimiser la récupération de FPDU chez le receveur MPA, une mise en œuvre de MPA/TCP optimisée utilise des règles de segmentation supplémentaires.

Pour fournir des performances optimales, une mise en œuvre optimisée du côté émetteur de MPA/TCP devrait être capable de :

* avec une EMSS assez grande pour contenir les FPDU, segmenter le flux TCP sortant de telle façon que le premier octet de chaque segment TCP commence avec une FPDU. Plusieurs FPDU peuvent être empaquetées dans un seul segment TCP pour autant qu'elles soient entièreement contenues dans le segment TCP.
* rapporter la EMSS courante de TCP à la couche de transmission MPA.

Il y a des exceptions à cette règle. Une fois qu'une ULPDU est fournie à MPA, l'envoyeur MPA/TCP la transmet ou fait échouer la connexion ; elle ne peut pas être répudiée. Par suite, durant les changements de MTU et d'EMSS, ou quand la taille de la fenêtre de réception (RWIN, *Receive Window*) de TCP devient trop petite, il peut être nécessaire d'envoyer des FPDU qui ne se conforment pas à la règle de segmentation ci-dessus.

Une solution de remplacement possible, mais moins désirable, est d'utiliser la fragmentation IP sur les FPDU acceptées pour traiter des réductions de MTU ou d'EMSS extrêmement petites.

Même quand l'alignement avec les segments TCP est perdu, l'envoyeur formate quand même la FPDU en accord avec le format de FPDU montré à la Figure 2.

Dans une retransmission, TCP ne préserve pas nécessairement les limites originales de la segmentation TCP. Cela peut conduire à la perte de l'alignement de FPDU et de la contenance au sein d'un segment TCP durant les retransmissions TCP. Un envoyeur MPA/TCP optimisé devrait essayer de préserver les limites originales de la segmentation TCP lors d'une retransmission.

A.2 Effets de la segmentation optimisée MPA/TCP

Un envoyeur MPA/TCP optimisé va remplir les segments TCP à l'EMSS avec une seule FPDU quand un message DDP est assez grand. Comme le message DDP ne peut pas tenir exactement dans les segments TCP, une "queue de message" se produit souvent résultant en une FPDU plus petite qu'un seul segment TCP. De plus, certains messages DDP peuvent être considérablement plus courts que l'EMSS. Si une petite FPDU est envoyée dans un seul segment TCP, le résultat est un "court" segment TCP. Les applications qui sont supposées trouver de forts avantages au placement direct de données incluent des applications fondées sur la transaction et des applications de débit. Les protocoles de demande/réponse envoient normalement une FPDU par segment TCP et ensuite attendent une réponse. Dans ces conditions, ces "courts" segments TCP sont un effet approprié et attendu de la segmentation.

Une autre possibilité est que l'application pourrait envoyer plusieurs messages (des FPDU) au même point d'extrémité avant d'attendre une réponse. Dans ce cas, la politique de segmentation tendrait à réduire la bande passante de connexion disponible en sous remplisssant les segments TCP.

Les mises en œuvre standard de TCP utilisent souvent l'algorithme de Nagle [RFC896] pour s'assurer que les segments sont

remplis à l'EMSS chaque fois que la latence d'aller-retour est assez grande pour que le flux de source puisse remplir complètement les segments avant que les accusés de réception (ACK) arrivent. L'algorithme fait cela en retardant la transmission des segments TCP jusqu'à ce que l'ULP puisse remplir un segment, ou jusqu'à ce qu'un ACK arrive du côté distant. L'algorithme permet donc de plus petits segments quand les latences sont plus courtes pour garder la latence de bout en bout de l'ULP à des niveaux raisonnables.

L'algorithme de Nagle n'est pas d'utilisation obligatoire [RFC1122].

Quand il est utilisé avec des piles MPA/TCP optimisées, Nagle et des algorithmes similaires peuvent résulter en la "mise en paquet" de plusieurs FPDU dans des segments TCP.

Si une "queue de message", de petits messages DDP, ou le début d'un plus grand message DDP sont disponibles, MPA peut mettre en paquet plusieurs FPDU dans des segments TCP. Quand c'est fait, les segments TCP peuvent être utilisés plus complètement, mais, du fait des contraintes de taille des FPDU, les segments ne peuvent pas être remplis à l'EMSS. Une MULPDU dynamique qui informe DDP de la taille de l'espace de segment TCP restant rend plus efficace le remplissage de segment TCP.

Noter que les receveurs MPA font plus de traitement sur un segment TCP qui contient plusieurs FPDU ; cela peut affecter les performances de certaines mises en œuvre de receveur.

Il appartient à l'ULP de décider si Nagle est utile avec DDP/MPA. Noter que beaucoup des applications supposées tirer parti de MPA/DDP préfèrent éviter les délais supplémentaires causés par Nagle. Dans ces scénarios, il est prévu qu'il y aura une opportunité minimale pour la mise en paquets à l'émetteur et que les receveurs peuvent choisir d'optimiser leurs performances pour ce comportement prévu.

Donc, l'application est supposée régler les paramètres TCP de telle façon qu'elle puisse faire un compromis entre la latence et l'efficacité du réseau. Les mises en œuvre devrait fournir une option de connexion qui désactive l'algorithme de Nagle pour MPA/TCP d'une façon similaire à celle dont l'option de prise TCP_NODELAY est fournie pour une interface de prises traditionnelle.

Quand la latence n'est pas critique, l'application est supposée laisser Nagle activé. Dans ce cas, la mise en œuvre de TCP peut mettre en paquets toutes les FPDU disponibles dans des segments TCP de façon que les segments soient remplis à l'EMSS. Si la quantité de données disponibles n'est pas suffisante pour remplir le segment TCP quand il est préparé pour la transmission, TCP peut envoyer le segment partiellement rempli, ou utiliser l'algorithme de Nagle pour attendre que l'ULP envoie plus de données.

A.3 Receveurs MPA/TCP optimisés

Quand une mise en œuvre de receveur MPA et le côté receveur d'une mise en œuvre de TCP à capacité MPA prennent en charge le traitement des ULPDU déclassées, la mise en œuvre de receveur TCP effectue les fonctions suivantes :

1) Elle passe les segments TCP entrants à MPA aussitôt qu'ils ont été reçus et validés, même si ils ne sont pas reçus en ordre. La couche TCP s'engage à garder chaque segment avant qu'il puisse être passé à MPA. Cela signifie que le segment doit avoir passé la validation d'intégrité de TCP, IP, et des couches inférieures (c'est-à-dire, la somme de contrôle) doit être dans la fenêtre de réception, doit faire partie de la même époque (si des horodatages sont utilisés pour le vérifier) et doit avoir passé toutes les autres vérifications requises par TCP.

   Ceci n'implique pas que les données doivent être complètement ordonnées avant utilisation. Une mise en œuvre peut accepter des segments en désordre, envoyer un accusé de réception sélectif à leur sujet (SACK, [RFC2018]) et les passer immédiatement à MPA, avant la réception des segments nécessaires pour boucher les trous. MPA attend pour utiliser ces segments qu'ils soient des FPDU complètes ou qu'ils puissent être combinés en FPDU complètes pour permettre de passer les ULPDU à DDP quand elles arrivent, indépendamment de leur ordre. DDP utilise les ULPDU passées pour "placer" les segments DDP (voir les détails dans la [RFC5041]).

   Comme MPA effectue un calcul de CRC et d'autres vérifications sur les FPDU reçues, la mise en œuvre de MPA/TCP s'assure que tout segment TCP qui duplique des données déjà reçues et traitées (comme cela peut arriver durant des essais TCP) n'écrase pas des FPDU déjà reçues et traitées. Cela évite la possibilité que des données dupliquées puissent corrompre des FPDU déjà validées.

2) La mise en œuvre fournit un mécanisme pour indiquer l'ordre des segments TCP dans lequel l'envoyeur les a transmis.

Un mécanisme possible pourrait être d'attacher le numéro de séquence TCP à chaque segment.

3) La mise en œuvre fournit aussi un mécanisme pour indiquer quand un certain segment TCP (et le flux TCP antérieur) est complet. Un mécanisme possible pourrait être d'utiliser le bord gauche de la fenêtre de réception TCP.

MPA utilise les indications d'ordre et d'achèvement pour informer DDP qu'une ULPDU est complète ; MPA livre la FPDU à DDP. DDP utilise les indications pour "livrer" ses messages au consommateur DDP (voir les détails dans la [RFC5041]).

DDP sur MPA utilise les deux mécanismes ci-dessus pour établir la sémantique de la livraison sur laquelle les consommateurs DDP s'accordent. Cette sémantique est décrite dans la [RFC5041]. Cela inclut l'exigence que le consommateur de DDP respecte la propriété des mémoires tampon avant le moment où DDP les livre au consommateur.

L'utilisation de SACK [RFC2018] améliore significativement l'utilisation et les performances du réseau et est donc recommandée. Quand elle est combinée avec la passation de segments dans le désordre à MPA et DDP, cela peut éviter une mise en mémoire tampon et un copiage significatifs des données reçues.

A.4 Resegmentation par boîtier de médiation et envoyeurs MPA/TCP non optimisés

Comme les envoyeurs MPA commencent souvent les FPDU sur une limite de segment TCP, une mise en œuvre receveuse optimisée de MPA/TCP peut être capable d'optimiser la réception de données de diverses façons.

Cependant, les receveurs MPA NE DOIVENT PAS dépendre de l'alignement de FPDU sur les limites de segment TCP.

Certains envoyeurs MPA peuvent être incapables de se conformer aux exigences de l'envoyeur parce que leur mise en œuvre de TCP n'est pas conçue dans l'esprit de MPA. Même pour des envoyeurs optimisés MPA/TCP, le réseau peut contenir des "boîtiers de médiation" qui modifient le flux TCP en changeant la segmentation. Ceci est généralement interopérable avec TCP et ses utilisateurs et MPA ne doit pas y faire exception.

La présence de marqueurs dans MPA (quand ils sont activés) permet à un receveur MPA/TCP optimisé de récupérer les FPDU en dépit de ces obstacles, bien qu'il puisse être nécessaire d'utiliser de la mémoire tampon supplémentaire chez le receveur pour le faire.

On mentionne ci-dessous pour rappel certains des cas qu'un receveur peut rencontrer :

* Une seule FPDU alignée et complète, en ordre ou en désordre : cela peut être passé à DDP sitôt que validé, et livré quand l'ordre est établi.

* Plusieurs FPDU dans un segment TCP, alignées et entièrement contenues, en ordre ou en désordre : cela peut être passé à DDP sitôt que validé, et livré quand l'ordre est établi.

* FPDU incomplète : le receveur devrait mettre en mémoire tampon jusqu'à ce que le reste de la FPDU arrive. Si le reste de la FPDU est déjà disponible, cela peut être passé à DDP aussitôt que validé, et livré quand l'ordre est établi.

* Début de FPDU non alignée : la FPDU partielle doit être combinée avec sa ou ses portions précédentes. Si les parties précédentes sont déjà disponibles, et si la FPDU entière est présente, cela peut être passé à DDP aussitôt que validé, et livré quand l'ordre est établi. Si la FPDU entière n'est pas disponible, le receveur devrait mettre en mémoire tampon jusqu'à ce que le reste de la FPDU arrive.

* Combinaisons de FPDU non alignées ou incomplètes (et potentiellement d'autres FPDU complètes) dans le même segment TCP : si une FPDU est présente entièrement, ou peut être complétée avec des portions déjà disponibles, elle peut être passée à DDP aussitôt que validée, et livrée quand l'ordre est établi.

A.5 Mise en œuvre de receveur

Mémoires tampon de réassemblage de couche transport & réseau :

L'utilisation de mémoires tampon de réassemblage (TCP ou de fragmentation IP) dépend de la mise en œuvre. Quand MPA est activé, les mémoires tampon de réassemblage sont nécessaires si des paquets arrivent en désordre et que les marqueurs ne sont pas activés. Les mémoires tampon sont aussi nécessaires si l'alignement de FPDU est perdu ou si une fragmentation

IP se produit. C'est parce que le segment entrant déclassé peut ne pas contenir assez d'informations pour que MPA traite toute la FPDU. Dans les cas où un boîtier de médiation qui resegmente est présent, ou lorsque l'envoyeur TCP n'est pas optimisé, la présence de marqueurs réduit significativement la quantité de mémoire tampon nécessaire.

La récupération de la fragmentation IP est transparente au consommateur MPA.

A.5.1 Mémoires tampon de réassemblage de couche réseau

La mise en œuvre de MPA/TCP devrait établir le bit IP Ne pas fragmenter à la couche IP. Donc, lors d'un changement de la MTU de chemin, les appareils intermédiaires éliminent le datagramme IP si il est trop grand et répondent avec un message ICMP qui dit à la source TCP que la MTU de chemin a changé. Cela cause l'émission par TCP de segments se conformant à la nouvelle taille de MTU de chemin. Donc, les fragments IP dans la plupart des conditions ne devraient jamais arriver chez le receveur. Mais c'est possible.

Il y a plusieurs options pour la mise en œuvre des mémoires tampon de réassemblage de couche réseau :

1. Éliminer tous les fragments IP, et répondre avec un message ICMP en accord avec la [RFC792] (fragmentation nécessaire et DF établi) pour dire à l'homologue distant de redimensionner son segment TCP.

2. Prendre en charge une mémoire tampon de réassemblage IP, mais de taille limitée (éventuellement de la même taille que la MTU de la liaison locale). Le nœud d'extrémité ne va normalement jamais annoncer une MTU de chemin supérieure à la MTU de liaison locale. Il est recommandé qu'un fragment IP éliminé cause la génération d'un message ICMP conformément à la [RFC0792].

3. Plusieurs mémoires tampon de réassemblage IP, de taille non limitée.

4. Prendre en charge une mémoire tampon de réassemblage IP pour le plus grand datagramme IP (64 kO).

5. Prendre en charge une grande mémoire tampon de réassemblage IP qui pourrait s'étendre sur plusieurs datagrammes IP.

Une mise en œuvre devrait prendre en charge au moins les options 2 ou 3, pour éviter d'éliminer les paquets qui ont traversé tout les tissu.

Il n'y a pas d'accusé de réception de bout en bout pour les mémoires tampon de réassemblage IP, de sorte qu'il n'y a pas de contrôle sur la mémoire tampon. Le seul accusé de réception de bout en bout est un ACK TCP, qui peut seulement se produire quand un datagramme IP complet est livré à TCP. À cause de cela, dans le pire des cas de scénarios pathologiques, la plus grande mémoire tampon de réassemblage IP est la fenêtre de réception TCP (pour mettre en mémoire tampon plusieurs datagrammes IP qui ont été tous fragmentés).

Noter que si l'homologue distant ne met pas en œuvre la segmentation des flux de données à réception de la réponse ICMP qui met à jour la MTU de chemin, il est possible que cela arrête la progression parce que l'homologue opposé va continuer de retransmettre en utilisant une taille de segment de transport trop grande. Ce scénario d'impasse n'est pas différent de si la MTU du tissu (pas la MTU du dernier bond) a été réduite après l'établissement de la connexion, et que le comportement du nœud distant n'est pas conforme à la [RFC1122].

A.5.2 Mémoires tampon de réassemblage TCP

Une mémoire tampon de réassemblage TCP est aussi nécessaire. Les mémoires tampon de réassemblage TCP sont nécessaires si l'alignement de FPDU est perdu quand on utilise TCP avec MPA ou quand la FPDU MPA s'étend sur plusieurs segments TCP. Des mémoires tampon sont aussi nécessaires si les marqueurs sont désactivés et que des paquets déclassés arrivent.

Comme la perte d'alignement de FPDU signifie souvent que les FPDU sont incomplètse, une mise en œuvre de MPA sur TCP doit avoir une mémoire tampon de réassemblage assez grande pour récupérer une FPDU qui est inférieure ou égale à la MTU de la liaison de rattachement local (ce devrait être la plus grande MTU de chemin possible annoncée parTCP). Si la MTU fait moins de 140 octets, une mémoire tampon d'au moins 140 octets est nécessaire pour prendre en charge la taille minimum de FPDU. Les 140 octets permettent une MULPDU minimum de 128 octets, 2 octets de bourrage, 2 de longueur d'ULPDU, 4 de CRC, et l'espace pour un possible marqueur. Comme d'habitude, de la mémoire tampon supplémentaire va probablement donner de meilleures performances.

Noter que si les segments TCP n'ont pas été mémorisés, il va être possible que l'algorithme de MPA arrive à une impasse. Si la MTU de chemin est réduite, l'alignement de FPDU exige que la source TCP resegmente le flux de données à la nouvelle MTU de chemin. La source MPA va détecter cette condition et réduire la taille de segment MPA, mais toutes les FPDU déjà envoyées à la source TCP vont être resegmentées et perdre l'alignement de FPDU. Si la destination ne prend pas en charge une mémoire tampon de réassemblage TCP, ces segments peuvent ne jamais être transmis et le protocole arrive à une impasse.

Quand une FPDU complète est reçue, le traitement se continue normalement.

Appendice B. Analyse du fonctionnement de MPA sur TCP

Cet Appendice est seulement pour information et NE fait PAS partie de la norme.

Cet Appendice est une analyse de MPA sur TCP et pourquoi il est utile pour intégrer MPA avec TCP (avec des modifications des mises en œuvre normales de TCP) pour réduire la mise en mémoire tampon globale des systèmes et leur frais généraux.

Un des buts généraux de MPA est de fournir suffisamment d'informations, quand il est combiné avec le protocole de placement direct des données [RFC5041], pour permettre le placement de charge utile déclassée de DDP dans la mémoire tampon finale du protocole de couche supérieure (ULP, *Upper Layer Protocol*). Noter que DDP sépare l'acte de placement des données dans une mémoire tampon d'ULP de celui de notifier à l'ULP que la mémoire tampon d'ULP est disponible à l'utilisation. Dans la terminologie de DDP, le premier est défini comme le "placement", et le dernier est défini comme une "livraison". MPA prend en charge la livraison dans l'ordre des données à l'ULP, incluant la prise en charge du placement direct des données dans la localisation finale de mémoire tampon d'ULP quand les segments TCP arrivent en désordre. Effectivement, le but est d'utiliser les mémoires tampon d'ULP pré affectées comme la mémoire tampon de réception TCP, où le réassemblage des unités de données de protocole (PDU, *Protocol Data Unit*) de l'ULP par TCP (avec MPA et DDP) est fait en place, dans la mémoire tampon d'ULP, sans copie des données.

Cet Appendice discute des avantages et inconvénients des modifications d'envoyeur TCP proposées par MPA :
1) que MPA préfère que l'envoyeur TCP fasse l'alignement d'en-tête, où un segment TCP devrait commencer avec une unité de données de protocole de tramage MPA (FPDU, *Framing Protocol Data Unit*) (si une charge utile est présente),
2) qu'il y a un nombre entier de FPDU dans un segment TCP (à condition que la MTU de chemin ne soit pas changée).

Cet Appendice conclut que les avantages d'adaptation de l'alignement de FPDU sont forts, sur la base d'une réduction drastique des exigences de mémoire tampon de réception TCP et d'un traitement simplifié à la récption. L'analyse montre aussi qu'il y a peu d'effets sur le comportement de TCP dans le réseau.

B.1 Hypothèses

B.1.1 MPA est mis en couches en dessous de DDP

MPA est une couche d'adaptation entre DDP et TCP. DDP exige la préservation des limites de segment DDP et un résumé de CRC32c couvrant l'en-tête et les données de DDP. MPA ajoute ces caractéristiques au flux TCP de sorte que DDP sur TCP a les mêmes propriétés de base que DDP sur SCTP.

B.1.2 MPA préserve le tramage de message DDP

MPA a été conçu comme une couche de tramage spécifiquement pour DDP et n'a pas été destinée à être une couche de trame d'utilisation générale pour tout autre ULP utilisant TCP.

Une couche de tramage permet aux ULP qui l'utilisent de recevoir des indications provenant de la couche de transport seulement quand des ULPDU complètes sont présentes. En tant que couche de tramage, MPA n'est pas instruit du contenu de la PDU DDP, seulement qu'il a reçu et, si nécessaire, réassemblé une PDU complète à livrer au DDP.

B.1.3 La taille de l'ULPDU passée à MPA est inférieure à l'EMSS dans des conditions normales

Pour rendre possibe la réception d'une PDU DDP complète sur chaque segment reçu, DDP passe à MPA une PDU qui ne fait pas plus que la EMSS du tissu sous-jacent. Chaque FPDU que crée MPA contient des informations suffisantes pour que le receveur place directement la charge utile d'ULP dans la localisation correcte de la mémoire tampon de réception correcte.

Des cas limites surviennent quand cette condition ne se réalise pas, mais n'ont pas besoin d'être traités en priorité.

B.1.4 Placement décalé mais livraison dans l'ordre

DDP reçoit des PDU DDP complètes de MPA. Chaque PDU DDP contient les informations nécessaires pour placer sa charge utile d'ULP directement dans la localisation correcte dans la mémoire de l'hôte.

Parce que chaque segment DDP est auto-descriptif, il est possible que les segments DDP reçus en désordre aient leur charge utile d'ULP placée immédiatement dans la mémoire tampon de réception d'ULP.

Il est garanti que la livraison des données à l'ULP va être dans l'ordre de l'envoi des données. DDP indique seulement la livraison des données à l'ULP après que TCP a accusé réception du flux d'octets complet.

B.2 Valeur de l'alignement de FPDU

Des optimisations de receveur significatives peuvent être réalisées quand l'alignement d'en-tête et des FPDU complètes sont le cas courant. Les optimisations permettent d'utiliser significativement moins de mémoire tampon chez le receveur et moins de calculs par FPDU. L'effet net est la capacité de construire un receveur "transflux" qui permet des solutions fondées sur TCP pour s'adapter à la 10G et au delà d'une façon économique. Les optimisations sont particulièrement pertinentes pour les mises en œuvre matérielles de receveurs qui traitent plusieurs couches de protocole -- couche de liaison des données (par exemple, Ethernet), couches réseau et transport (par exemple, TCP/IP) et même certains ULP par dessus TCP (par exemple, MPA/DDP). Avec l'augmentation de la vitesse des réseaux, il y a un désir croissant d'utiiser un receveur fondé sur le matériel afin de réaliser une solution efficace de hautes performances.

Un receveur TCP, dans les pires conditions, doit allouer des mémoires tampon (BufferSizeTCP) dont les capacités sont fonction du produit de la bande passante et du délai. Donc :

BufferSizeTCP = K * bande passante [octets/seconde] * Délai [secondes].

Où "bande passante" est la bande passante de bout en bout de la connexion, "délai" est le délai d'aller-retour de la connexion, et K est une constante dépendante de la mise en œuvre.

Donc, BufferSizeTCP s'adapte avec la bande passante de bout en bout (10 fois plus de mémoire tampon pour une augmentation de dix fois de la bande passante de bout en bout). Comme cette approche de la mise en mémoire tampon peut mal s'adapter pour certaines mises en œuvre de matériel ou de logiciel, plusieurs approches permettent une réduction de la quantité de mémoire tampon requise pour les communications TCP à haut débit.

L'approche de MPA/DDP est de permettre à la mémoire tampon de l'ULP d'être utilisée comme mémoire tampon de réception TCP. Si l'application pré-alloue une quantité suffisante de mémoire tampon, et si chaque segment TCP a des informations suffisantes pour placer la charge utile dans la bonne mémoire tampon d'application, quand un segment TCP arrive déclassé, il pourrait éventuellement être placé directement dans la mémoire tampon d'ULP. Cependant, le placement peut seulement être fait quand une FPDU complète avec les informations de placement est disponible au receveur, et quand le contenu de la FPDU contient suffisamment d'informations pour placer les données dans la mémoire tampon d'ULP correcte (par exemple, quand un en-tête DDP est disponible).

Pour le cas où la FPDU n'est pas alignée avec le segment TCP, il peut falloir, en moyenne, 2 segments TCP pour assembler une FPDU. Donc, le receveur doit allouer BufferSizeNAF (taille de mémoire tampon, FPDU non alignée) octets:

BufferSizeNAF = K1* EMSS * nombre_de_connexions + K2 * EMSS

Où K1 et K2 sont des constantes qui dépendent de la mise en œuvre et EMSS est la taille effective maximum de segment.

Par exemple, une liaison à 1 GO/s avec 10 000 connexions et une EMSS de 1500 O va exiger 15 MO de mémoire. Souvent

le nombre de connections utilisé s'adapte à la vitesse du réseau, aggravant la situation pour de plus grandes vitesses.

L'alignement de FPDU va permettre au receveur d'allouer BufferSizeAF (taille de mémoire tampon, FPDU alignée) octets :

BufferSizeAF = K2 * EMSS

pour les mêmes conditions. Un receveur de FPDU alignée peut exiger de la mémoire dans la gamme de ~100s de kO -- ce qui est faisable pour une mémoire à puce et permet une conception "transflux" dans laquelle les données s'écoulent à travers la carte d'interface réseau (NIC, *Network Interface Card*) et sont placées directement dans la mémoire tampon de destination. En supposant que la plupart des connexions prennent en charge l'alignement de FPDU, la mémoire tampon du receveur ne s'adapte plus au nombre de connexions.

Des optimisations supplémentaires peuvent être réalisées dans un sous système équilibré d'entrée/sortie -- où l'interface système du contrôleur du réseau fournit une ample bande passante comparée à celle du réseau. Depuis presque vingt ans cela a été le cas et la tendance est supposée se continuer. Alors que les vitesses d'Ethernet se sont multipliées par 1000 (de 10 mégabit/s à 10 gigabit/s) la bande passante d'un bus entrée/sortie des volumes d'architectures de CPU s'est adaptée de ~2 MO/s à ~2 GO/s (du bus PC-XT au DDR PCI-X). Dans ces conditions, l'approche de l'alignement de FPDU permet à la taille de mémoire tampon pour FPDU alignée (BufferSizeAF) d'être indifférente à la vitesse du réseau. Elle est principalement fonction du temps de traitement local d'une trame donnée.

Donc, quand l'approche de l'alignement de FPDU est utilisée, la mémoire tampon de réception est supposée s'adapter en douceur (c'est-à-dire, moins qu'une adaptation linéaire) lorsque la vitesse du réseau augmente.

B.2.1 Impact du manque d'alignement de FPDU sur la charge de calcul et la complexité chez le receveur

Le receveur doit effectuer le traitement IP et TCP, puis effectuer les vérifications de CRC de FPDU, avant de pouvoir faire confiance aux informations de placement d'en-tête de FPDU. Pour la simplicité de la description, l'hypothèse est qu'une FPDU est portée dans pas plus de deux segments TCP. En réalité, sans alignement de FPDU, une FPDU peut être portée dans plus de deux segments TCP (par exemple, si la MTU de chemin a été réduite).

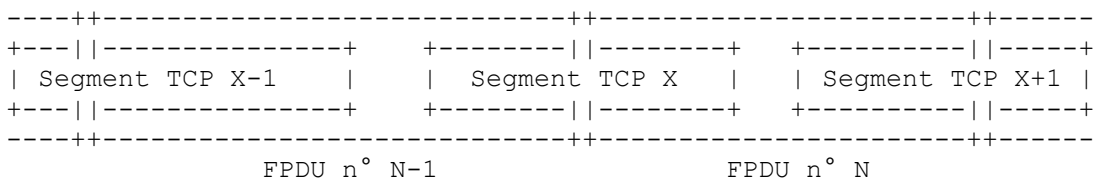

Figure 12 : FPDU non alignée placée librement dans un flux d'octets TCP

L'algorithme de receveur pour le traitement des segments TCP (par exemple, le segment TCP n° X à la Figure 12) portant des FPDU non alignées (en ordre ou en désordre) inclut :

Traitement de couche de liaison des données (trame entière) -- incluant normalement un calcul de CRC.

1. Traitement de couche réseau (en supposant qu'il n'y a pas de fragment IP, la trame de couche liaison des données entière contient un datagramme IP. Les fragments IP devraient être réassemblés dans une mémoire tampon locale. Ceci n'est pas un but d'optimisation de performances.)

2. Traitement de couche Transport -- traitement de protocole TCP, vérification d'en-tête et de somme de contrôle.
   a. Classer les segments TCP entrants en utilisant le quintuplet (source IP, destination IP, accès de source TCP, accès de destination TCP, protocole).

3. Trouver les limites de message de la FPDU.
   a. Obtenir les informations d'état MPA pour la connexion. Si le segment TCP est dans l'ordre, utiliser les informations d'état MPA gérées par le receveur pour calculer où se termine le précédent message de FPDU (n° N-1) dans le segment TCP X courant. (Précédemment, quand le receveur MPA a traité la première partie de la FPDU n° N-1, il a calculé le nombre d'octets restants pour compléter la FPDU n° N-1 en utilisant le champ Longueur MPA).

   Obtenir le CRC partiel CRC mémorisé pour la FPDU n° N-1.
   Compléter le calcul de CRC pour les données de la FPDU n° N-1 (première portion du segment TCP n° X).
   Vérifier le calcul de CRC pour la FPDU n° N-1.

Si il n'y a pas d'erreur de CRC de FPDU, le placement est permis.
Localiser la mémoire tampon locale pour la première portion de la FPDUn° N-1. Copier les données de (mémoire tampon locale de la première portion de la FPDU n° N-1, adresse de mémoire tampon d'hôte, longueur).
Calculer l'adresse de la mémoire tampon d'hôte pour la seconde portion de la FPDU n° N-1.
Copier les données de (mémoire tampon locale de la seconde portion de la FPDU n° N- 1, adresse de mémoire tampon d'hôte pour la seconde portion, longueur).
Calculer le décalage d'octets dans le segment TCP pour la prochaine FPDU n° N.
Commencer le calcul du CRC pour les données disponibles de la FPDU n° N.
Mémoriser le résultat du CRC partiel pour la FPDU n° N.
Mémoriser l'adresse de mémoire tampon locale de la première portion de la FPDU n° N.
Aucune autre action n'est possible sur la FPDU n° N, avant qu'elle soit complètement reçue.
Si le segment TCP est en désordre, le receveur doit mettre en mémoire tampon les données jusqu'à ce qu'au moins une FPDU complète soit reçue. Normalement, la mise en mémoire tampon pour plus d'un segment TCP par connexion est exigée. L'utilisation de marqueurs fondés sur MPA pour calculer où sont les limites de FPDU.
Quand une FPDU complète est disponible, une procédure similaire à celle de l'algorithme dans l'ordre ci-dessus est utilisée. Il y a cependant une complexité supplémentaire parce que quand le segment manquant arrive, ce segment TCP doit être passé par le moteur de CRC après que le CRC est calculé pour le segment manquant.

Si on suppose l'alignement de FPDU, le diagramme suivant et l'algorithme donné en dessous s'appliquent. Noter que quand on utilise MPA, le receveur est supposé détecter activement la présence ou la perte de l'alignement de FPDU pour chaque segment TCP reçu.

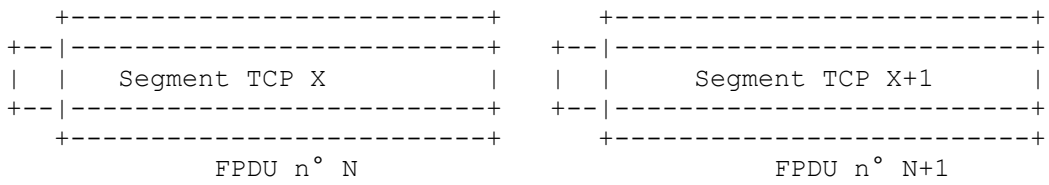

Figure 13 : FPDU alignée placée immédiatement après l'en-tête TCP

L'algorithme de receveur pour les trames de FPDU alignée (en ordre ou en désordre) inclut :

1) Traitement de couche de liaison des données (toute la trame) -- incluant normalement un calcul de CRC.

2) Traitement de couche réseau (en supposant que ce n'est pas un fragment IP, la trame de couche de liaison des données entière contient un datagramme IP. Les fragments IP devraient être réassemblés dans une mémoire tampon locale. Ceci n'est pas un but d'optimisation de performances.)

3) Traitement de couche Transport -- traitement de protocole TCP, vérifications d'en-tête et de somme de contrôle.
   a. Classer le segment TCP entrant en utilisant le quintuplet (source IP, destination IP, accès de source TCP, accès de destination TCP, protocole).

4) Vérification de l'alignement d'en-tête (décrit en détails à la Section 6). En supposant l'alignement d'en-tête pour le reste de l'algorithme ci-dessous.
   a. Si l'en-tête n'est pas aligné, voir l'algorithme défini au paragraphe précédent.

5) Si le segment TCP est en ordre ou en désordre, l'en-tête MPA est au début de la charge utile TCP courante. Obtenir la longueur de la FPDU dans l'en-tête de la FPDU.

6) Calculer le CRC sur la FPDU.

7) Vérifier le calcul de CRC pour la FPDU n° N.

8) Si il n'y a pas d'erreur de CRC de la FPDU, le placement est permis.

9) Copier les données de (segment TCP n° X, adresse de mémoire tampon d'hôte, longueur).

10) Reboucler au n° 5 jusqu'à ce que toutes les FPDU dans le segment TCP soient consommées afin de traiter la mise en paquet de FPDU.

Note de mise en œuvre : dans les deux cas, le receveur doit classer le segment TCP entrant et l'associer à un des flux qu'il maintient. Dans le cas où il n'y a pas d'alignement de la FPDU, le receveur est forcé de classer le trafic entrant avant de pouvoir calculer le CRC de la FPDU. En cas d'alignement de la FPDU, l'ordre des opérations est laissé à la mise en œuvre.

L'algorithmede receveur de FPDU alignée est significativement plus simple. Il n'est pas besoin de mettre en mémoire tampon locale des portions de FPDU. Les informations d'état d'accès sont aussi substantiellement simplifiées --le cas normal n'exige pas de restituer des informations pour trouver où commence et finit une FPDU ou restituer un CRC partiel avant que le calcul de CRC puisse commencer. Cela évite d'ajouter des latences internes, ayant plusieurs passages de données à travers la machine de CRC, ou de programmer plusieurs commandes pour déplacer les données dans la mémoire tampon de l'hôte.

L'approche de la FPDU alignée est utile pour la réception en ordre et en désordre. Le receveur peut utiliser les mêmes mécanismes pour la mémorisation de données dans les deux cas, et a seulement besoin de tenir compte de quand tous les segments TCP sont arrivés pour activer la livraison. L'alignement d'en-tête, avec la forte probabilité qu'au moins une FPDU complète soit trouvée avec chaque segment TCP, permet au receveur d'effectuer le placement de données pour les segments TCP en désordre sans avoir besoin d'une mise en mémoire tampon intermédiaire. Essentiellement, la mémoire tampon de réception TCP a été éliminée et le réassemblage TCP est fait à la place au sein de la mémoire tampon d'ULP.

Au cas où l'alignement de FPDU n'est pas trouvé, le receveur devrait suivre l'algorithme pour la réception de FPDU non alignée, qui peut être plus lent et moins efficace.

B.2.2 Effets de l'alignement de FPDU sur le protocole réseau TCP

Dans une mise en œuvre optimisée de MPATCP, TCP expose son EMSS à MPA. MPA utilise l'EMSS pour calculer sa MULPDU, qu'il expose ensuite à DDP, son ULP. DDP utilise la MULPDU pour segmenter sa charge utile afin que chaque FPDU envoyée par MPA tienne complètement dans un segment TCP. Cela n'a pas d'impact sur le protocole réseau, et exposer ces informations est déjà pris en charge sur de nombreuses mises en œuvre de TCP, incluant toutes les nuances modernes de réseautage BSD, par l'option de prise TCP_MAXSEG.

Dans le cas le plus courant, les messages d'ULP (c'est-à-dire, DDP sur MPA) fournis à la couche TCP sont segmentés à la taille de la MULPDU. Il est suppposé que la taille du message d'ULP est limitée par la MULPDU, de sorte qu'un seul message d'ULP peut être encapsulé dans un seul segment TCP. Donc, dans le cas courant, il n'y a pas d'augmentation du nombre de segments TCP émis. Pour de plus petits messages d'ULP, l'envoyeur peut aussi appliquer la mise en paquet, c'est-à-dire, l'envoyeur met en paquets autant de FPDU complètes que possible dans un segment TCP. L'exigence de toujours avoir une FPDU complète peut augmenter le nombre de segments TCP émis. Normalement, la taille d'un message d'ULP varie de quelques octets à plusieurs EMSS (par exemple, 64 k octets). Dans certains cas, l'ULP peut poster plus d'un message à la fois pour la transmission, donnant à l'envoyeur l'opportunité de mettre en paquets. Dans le cas où plus d'une FPDU est disponible pour la transmission et où les FPDU sont encapsulées dans un segment TCP et qu'il n'y a pas de place dans le segment TCP pour inclure la prochaine FPDU complète, un autre segment TCP est envoyé. Dans ce cas limite, certains des segments TCP ne sont pas de taille complète. Dans le pire des scénarios, l'ULP peut choisir une taille de FPDU qui est EMSS/2 +1 et a plusieurs messages disponibles à transmettre. Pour ce mauvais choix de taille de FPDU, la taille moyenne de segment TCP est donc d'environ 1/2 de la EMSS et le nombre de segments TCP émis approche 2x ce qui est possible sans l'exigence d'encapsuler un nombre entier de FPDU complètes dans chaque segment TCP. C'est une situation dynamique qui ne dure que pendant le temps où l'ULP envoyeur a plusieurs messages non optimaux à transmettre et cela ne cause qu'un impact mineur sur l'utilisation du réseau.

Cependant, il n'est pas prévu qu'exiger l'alignement de FPDU ait un impact mesurable sur le comportement du réseau de la plupart des applications. Les applications de débit avec de grandes entrées/sorties sont supposées tirer pleinement parti de la EMSS. Une autre classe d'applications avec de nombreuses petites mémoires tampon en instance (par rapport à l'EMSS) est supposée utiliser la mise en paquets quand elle est applicable. Les applications en mode transaction sont aussi optimales.

La retransmission TCP est un autre domaine qui peut affecter le comportement de l'envoyeur. TCP prend en charge la retransmission du segment exact, originellement transmis (voir les paragraphes 2.6 et 3.7 de la [RFC793] (sous "Gestion de la fenêtre") et le paragraphe 4.2.2.15 de la [RFC1122]). Dans le cas improbable où une partie du segment original aurait été reçue et acquittée par l'homologue distant (par exemple, un boîtier de médiation qui resegmente, comme exposé à l'Appendice A.4, "Resegmentation par boîtier de médiation et envoyeurs MPA/TCP non optimisés") une meilleure utilisation de la bande passante disponible serait possible en retransmettant seulement les octets manquants. Si un MPA/TCP optimisé retransmet des FPDU complètes, il peut y avoir des pertes marginales de bande passante.

Un autre domaine où un changement du nombre de segments TCP peut avoir un impact est celui du démarrage lent et de l'évitement d'encombrement. L'augmentation exponentielle du démarrage lent est mesurée en segments par seconde, car l'algorithme se concentre sur les frais généraux par segment à la source pour l'encombrement qui résulte finalement en segments éliminés. La croissance exponentielle de la bande passante dans le démarrage lent pour MPA/TCP optimisé est similaire à celle de toute mise en œuvre de TCP. L'évitement d'encombrement permet une croissance linéaire de la bande passante disponible quand on récupère après un abandon de paquet. Comme pour l'analyse du démarrage lent, le MPA/TCP optimisé ne change pas le comportement de l'algorithme. Donc, l'opposition taille moyenne du segment contre EMSS n'est pas un facteur majeur dans l'établissement de la croissance de la bande passants pour un envoyeur. Le démarrage lent et l'évitement d'encombrement pour un MPA/TCP optimisé vont tous deux se comporter de façon similaire à celle de tout envoyeur TCP et permettent à un MPA/TCP optimisé de bénéficier des limites de performances théoriques des algorithmes.

En résumé, les messages d'ULP générés chez l'envoyeur (par exemple, la quantité de messages groupés pour chaque demande de transmission) et la distribution des tailles de message a l'impact le plus significatif sur le nombre de segments TCP émis. Le plus mauvais effet pour certains ULP (avec une taille moyenne de message de EMSS/2+1 à EMSS) est limité par une augmentation de jusqu'à deux fois le nombre de segments TCP et accusés de réception. En réalité, l'effet est supposé être marginal.

Appendice C. Interopérabilité de mise en œuvre IETF avec les protocoles du consortium RDMA

Cet Appendice est seulement pour information et NE fait PAS partie de la norme.

Cet Appendice traite des méthodes pour faire interopérer les mises en œuvre de MPA avec les deux versions des protocoles, de l'IETF, et du Consortium RDMA.

Le Consortium RDMA a créé les premières spécifications des protocoles MPA/DDP/RDMA, et certains fabricants ont créé des mises en œuvre de ces protocoles avant que les versions de l'IETF soient finalisées. Ces protocoles sont très similaires à ceux des versions de l'IETF, rendant possible que des mises en œuvre soient créées ou modifiées pour prendre en charge l'un et l'autre ensemble de spécifications.

Pour ceux que cela intéresse, les documents de protocole du Consortium RDMA ([RDMA-MPA], [RDMA-DDP], et [RDMA-RDMAC]) peuvent être obtenus à http://www.rdmaconsortium.org/home.

Dans cette section, les mises en œuvre de MPA/DDP/RDMA qui se conforment aux spécifications du Consortium RDMA sont appelées RNIC RDMAC. Les mises en œuvre de MPA/DDP/RDMA qui se conforment aux RFC de l'IETF sont appelées des RNIC IETF.

Sans l'échange de trames de demande/réponse MPA, il n'y a pas de mécanisme standard pour permettre aux RNIC RDMAC d'interopérer avec les RNIC IETF. Même si un ULP utilise un accès bien connu pour démarrer un RNIC IETF immédiatement en mode RDMA (c'est-à-dire, sans échanger les messages de demande/réponse MPA) il n'y a pas de raison de croire qu'un RNIC IETF va interopérer avec un RNIC RDMAC à cause des differences dans le numéro de version dans les en-têtes DDP et RDMAP sur le réseau.

Donc, l'ULP ou autre entité de prise en charge au RNIC RDMAC doit mettre en œuvre des trames de demande/réponse MPA au nom du RNIC afin de négocier les paramètres de connexion. Les paragraphes qui suivent décrivent les résultats de l'échange de trames de demande/réponse MPA avant la conversion du mode de flux directs en mode RDMA.

C.1 Paramètres négociés

Trois types de RNIC sont considérés :

RNIC RDMAC mis à niveau : un RNIC mettant en œuvre les protocoles RDMAC qui ont un ULP ou autre entité de prise en charge qui échange les trames de demande/réponse MPA en mode de flux directs avant la conversion en mode RDMA.

RNIC IETF non permissif : un RNIC mettant en œuvre les protocoles de l'IETF qui n'est pas capable de mettre en œuvre les protocoles RDMAC. Un tel RNIC peut seulement interopérer avec d'autres RNIC IETF.

RNIC IETF permissif : un RNIC mettant en œuvre les protocoles de l'IETF qui est capable de mettre en œuvre les

protocoles RDMAC connexion par connexion.

Le RNIC IETF permissif est recommandé pour les mises en œuvres qui veulent un maximum d'interopérabilité avec les autres mises en œuvre de RNIC.

Les valeurs utilisées par ces trois types de RNIC pour les versions de MPA, DDP, et RDMAP ainsi que les marqueurs MPA et les CRC sont résumés à la Figure 14.

Type de RNIC	Version de DDP/RDMAP	Révision de MPA	Marqueurs MPA	CRC MPA
RDMAC	0	0	1	1
Ne permet pas l'IETF	1	1	0 ou 1	0 ou 1
Permet l'IETF	1 ou 0	1 ou 0	0 ou 1	0 ou 1

activé = 1, désactivé = 0.

Figure 14 : Paramètres de connexion pour les types de RNIC pour les marqueurs et CRC MPA

On suppose qu'il n'y a pas de mélange de versions permis entre MPA, DDP, et RDMAP. Le RNIC génère les protocoles RDMAC sur le réseau (la version est zéro) ou utilise les protocoles de l'IETF (la version est un).
Durant l'échange des trames de demande/réponse MPA, chaque homologue fournit sa révision de MPA, sa préférence de marqueur (M = 0, désactivé, 1, activé) et sa préférence de CRC. La révision MPA fournie dans la trame de demande MPA et dans la trame de réponse MPA peut différer.
D'après les informations des trames demande/réponse MPA, chaque côté règle le champ Version (V = 0, RDMAC, 1 = IETF) du protocole DDP/RDMAP ainsi que l'état des marqueurs pour chaque demie connexion. Entre DDP et RDMAP, aucun mélange de versions n'est permis. De plus, la version DDP et RDMAP DOIT être identique dans les deux sens. Le RNIC génère les protocoles RDMAC sur le réseau (version zéro) ou utilise les protocoles de l'IETF (version un).
Dans les paragraphes qui suivent, les figures ne discutent pas de la négociation de CRC parce que il n'y a pas de problème d'interoperabilité pour les CRC. Comme le RNIC RDMAC va toujours demander l'utilisation du CRC, conformément à la spécification MPA de l'IETF, les deux homologues DOIVENT alors générer et vérifier les CRC.

C.2 RNIC RDMAC et RNIC ne permettant pas l'IETF

La Figure 15 montre qu'un RNIC IETF non permissif ne peut pas interopérer avec un RNIC RDMAC, en dépit du fait que les deux homologues échangent des trames de demande/réponse MPA. Pour un RNIC IETF non permissif, la négociation MPA n'a pas d'effet sur la version de DDP/RDMAP et il n'est pas capable d'interopérer avec le RNIC RDMAC.
Les rangées dans la figure montrent l'état du champ Marqueur dans la trame de demande MPA envoyée par l'initiateur MPA. Les colonnes montrent l'état du champ Marqueur dans la trame de réponse MPA envoyée par le répondeur MPA. Chaque type de RNIC est montré comme Initiateur et Répondeur. Les résultats de la connexion sont montrés dans le coin inférieur droit, à l'intersection des différents types de RNIC, où V=0 est la version RDMAC DDP/RDMAP, V=1 est la version IETF DDP/RDMAC, M=0 signifie que les marqueurs MPA sont désactivés, et M=1 signifie que les marqueurs MPA sont activés. L'état de marqueur négocié est montré par X/Y, pour la direction de réception de l'initiateur/répondeur.

Mode de connexion MPA			Répondeur MPA		
	Type de RNIC		RDMAC	IETF non permissif	
		Marqueur	M=1	M=0	M=1
Initiateur MPA	RDMAC	M=1	V=0 M=1/1	clôt	clôt
	IETF non permis.	M=0	clôt	V=1 M=0/0	V=1 M=0/1
		M=1 - --	clôt	V=1 M=1/0	V=1 M=1/1

Figure 15 : Négociation MPA entre un RNIC RDMAC et un RNIC ne permettant pas l'IETF

C.2.1 Initiateur RNIC RDMA

Si le RNIC RDMAC est l'initiateur MPA, son ULP envoie une trame de demande MPA avec le champ Rev réglé à zéro et les bits M et C réglés à un. Parce que le RNIC IETF non permissif ne peut pas dégrader dynamiquement le numéro de version qu'il utilise pour DDP et RDMAP, il va envoyer une trame de réponse MPA avec le champ Rev égal à un et ensuite clore en douceur la connexion.

C.2.2 Initiateur RNIC ne permettant pas l'IETF

Si le RNIC IETF non permissif est l'initiateur MPA, il envoie une trame de demande MPA avec le champ Rev égal à un. L'ULP ou l'entité de prise en charge du RNIC RDMAC répond avec une trame de réponse MPA qui a le champ Rev égal à zéro et le bit M réglé à un. Le RNIC IETF non permissif va clore en douceur la connexion après avoir lu le champ Rev incompatible dans la trame de réponse MPA.

C.2.3 RNIC RDMAC et RNIC permettant l'IETF

La Figure 16 montre qu'un RNIC IETF permissif peut interopérer avec un RNIC RDMAC sans considération de sa préférence de marqueur. La figure utilise le même format que montré avec le RNIC IETF non permissif.

Mode de connexion MPA		Répondeur MPA			
	Type de RNIC	RDMAC	IETF permissif		
		Marqueur	M=1	M=0	M=1
Initiateur MPA	RDMAC	M=1	V=0 M=1/1	N/A	V=0 M=1/1
	IETF	M=0	V=0 M=1/1	V=1 M=0/0	V=1 M=0/1
	permis.	M=1 - --	V=0 M=1/1	V=1 M=1/0	V=1 M=1/1

Figure 16 : Négociation MPA entre un RNIC RDMAC et un RNIC permettant l'IETF

Un vrai RNIC IETF permissif va reconnaître un RNIC RDMAC d'après le champ Rev des trames de demande/réponse MPA et ajuster alors son état de marqueur de réception et la version DDP/RDMAP pour s'accommoder du RNIC RDMAC. Par suite, comme répondeur MPA, le RNIC IETF permissif ne va jamais retourner une trame de réponse MPA avec le bit M réglé à zéro. Ce cas est montré comme non applicable (N/A) à la Figure 16.

C.2.4 Initiateur RNIC RDMAC

Quand le RNIC RDMAC est l'initiateur MPA, son ULP ou autre entité de support prépare un message de demande MPA et règle la révision à zéro et les bits M et C à un.

Le répondeur IETF permissif reçoit le message de demande MPA et vérifie le champ Révision. Comme il est capable de générer des en-têtes RDMAC DDP/RDMAP, il envoie un message de réponse MPA avec la révision réglée à zéro et les bits M et C établis à un. Le répondeur doit informer son ULP qu'il génère la version zéro des messages DDP/RDMAP.

C.2.5 Initiateur RNIC permettant l'IETF

Si le RNIC IETF permissif est l'initiateur MPA, il prépare la trame de demande MPA en réglant le champ Rev à un. Sans considération de la valeur du bit M dans la trame de demande MPA, l'ULP ou autre entité de support pour le RNIC RDMAC va créer une trame de réponse MPA avec Rev égal à zero et le bit M établi à un.

Quand l'initiateur lit le champ Rev de la trame de réponse MPA et trouve que son homologue est un RNIC RDMAC, il doit

informer son ULP qu'il devrait générer la version zéro de messages DDP/RDMAP et activer les marqueurs et CRC MPA.

C.3 RNIC ne permettant pas l'IETF et RNIC permettant l'IETF

Pour être complet, la Figure 17 montre les résultats de la négociation MPA entre un RNIC IETF non permissif et un RNIC IETF permissif. Le point important de cette figure est qu'un RNIC IETF ne peut pas détecter si son homologue est un RNIC permissif ou non permissif.

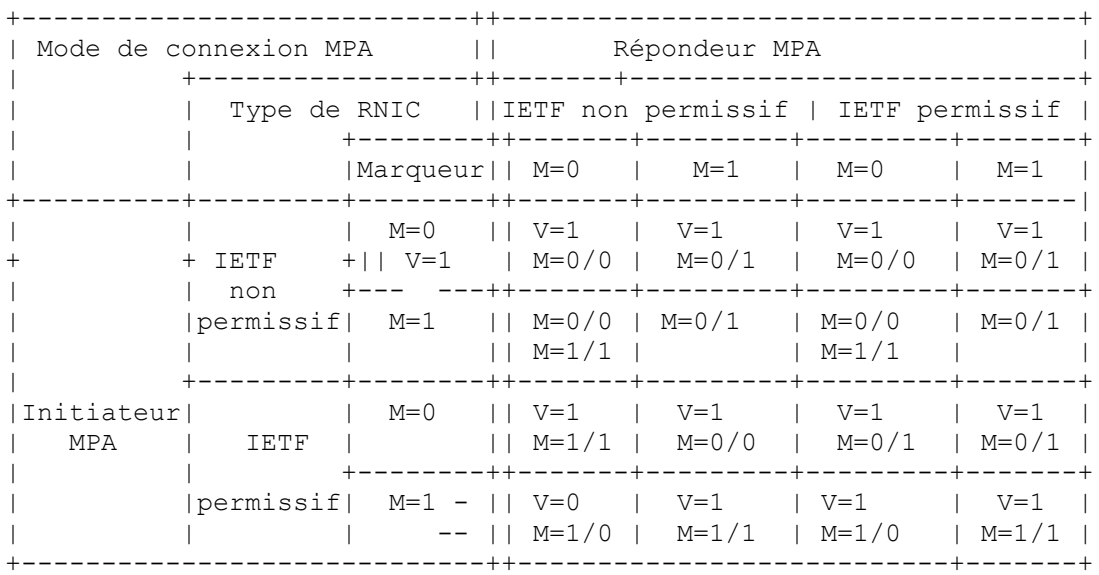

Figure 17 : négociation MPA entre un RNIC ne permettant pas l'IETF et un RNIC permettant l'IETF

Références normatives

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981. (*Remplacée par* RFC9293)

[RFC1191] J. Mogul et S. Deering, "Découverte de la MTU de chemin", novembre 1990.

[RFC2018] M. Mathis et autres, "Options d'accusé de réception sélectif sur TCP", octobre 1996. (*Remplace* RFC1072) *(P.S.)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC3720] J. Satran et autres, "Interface Internet des systèmes de petits ordinateurs (iSCSI)", avril 2004. *(Remplacée par* RFC7143)

[RFC3723] B. Aboba et autres, "Protocoles de sécurisation de mémorisation de blocs sur IP", avril 2004. *(P.S.)*

[RFC5042] J. Pinkerton, E. Deleganes, "Sécurité du protocole de placement direct des données (DDP) / protocole d'accès direct à une mémoire distante (RDMAP)", octobre 2007. *(P.S. ; MàJ par* RFC7146*)*

Références pour information

[CRCTCP] Stone J., Partridge, C., "When the CRC and TCP checksum disagree", ACM Sigcomm, septembre 2000.

[DAT-API] DAT Collaborative, "kDAPL (Kernel Direct Access Programming Library) and uDAPL (User Direct Access Programming Library)", http://www.datcollaborative.org.

[IT-API] The Open Group, "Interconnect Transport API (IT-API)" Version 2.1, http://www.opengroup.org.

[RDMA-DDP] RDMA Consortium, "Direct Data Placement over Reliable Transports (Version 1.0)", octobre 2002, <http://www.rdmaconsortium.org/home/draft-shah-iwarp-ddp-v1.0.pdf>.

[RDMA-MPA] RDMA Consortium, "Marker PDU Aligned Framing for TCP Specification (Version 1.0)", octobre 2002, <http://www.rdmaconsortium.org/home/draft-culley-iwarp-mpa-v1.0.pdf>.

[RDMA-RDMAC] "An RDMA Protocol Specification (Version 1.0)", RDMA Consortium, octobre 2002, <http://www.rdmaconsortium.org/home/draft-recio-iwarp-rdmac-v1.0.pdf>.

[RFC0792] J. Postel, "Protocole du message de contrôle Internet – Spécification du protocole du programme Internet DARPA", STD 5, septembre 1981. *(MàJ par la* RFC6633)

[RFC0896] J. Nagle, "Contrôle de l'encombrement dans l'inter-réseau IP/TCP", janvier 1984. *(Historique)*

[RFC1122] R. Braden, "Exigences pour les hôtes Internet – couches de communication", STD 3, octobre 1989. *(MàJ par* RFC6633, 8029, 9293)

[RFC4296] S. Bailey, T. Talpey, "L'architecture de placement de données directes (DDP) et d'accès direct en mémoire distante (RDMA) avec les protocoles de l'Internet", décembre 2005. *(Information)*

[RFC4297] A. Romanow et autres, "Position du problème de l'accès direct en mémoire distante (RDMA) sur IP", décembre 2005. *(Info.)*

[RFC4301] S. Kent et K. Seo, "Architecture de sécurité pour le protocole Internet", décembre 2005. *(P.S.) (Remplace la* RFC2401)

[RFC4960] R. Stewart, éd., "Protocole de transmission de commandes de flux (SCTP)", septembre 2007. (*Remplace* RFC2960, RFC3309 ; *P.S. ; Remplacée par* RFC9260)

[RFC5040] R. Recio et autres, "Spécification d'un protocole d'accès direct à une mémoire distante", octobre 2007. *(P.S. ; MàJ par* RFC7146 *)*

[RFC5041] H. Shah et autres, "Placement direct des données sur transports fiables", octobre 2007. *(P.S. ; MàJ par RFC 7146)*

[RFC5045] C. Bestler et autres, "Applicabilité du protocole d'accès direct à une mémoire distante (RDMA) et du placement direct des données (DDP)", octobre 2007. *(Information)*

[RFC5046] M. Ko et autres, "Extensions pour l'accès direct à une mémoire distante (RDMA) à l'interface système de petit ordinateur à l'Internet (iSCSI)", octobre 2007. *(P.S. ; Remplacée par* RFC7145)

[RFC5056] N. Williams, "Sur l'utilisation des liens de canaux pour sécuriser les canaux", novembre 2007. *(P.S.)*

[VERBS-RMDA] "RDMA Protocol Verbs Specification", RDMA Consortium standard, avril 2003, <http://www.rdmaconsortium.org/home/draft-hilland-iwarp-verbs-v1.0-RDMAC.pdf>.

Contributeurs

Dwight Barron, Hewlett-Packard Company ; mél : dwight.barron@hp.com
Jeff Chase, Duke University ; mél : chase@cs.duke.edu
Ted Compton, EMC Corporation ; mél : compton_ted@emc.com
Dave Garcia ; mél : Dave.Garcia@StanfordAlumni.org
Hari Ghadia, Gen10 Technology, Inc.; mél : hghadia@gen10technology.com
Howard C. Herbert, Intel Corporation ; mél : howard.c.herbert@intel.com

Jeff Hilland, Hewlett-Packard Company ; mél : jeff.hilland@hp.com
Mike Ko, IBM ; mél : mako@us.ibm.com
Mike Krause, Hewlett-Packard Corporation, ; mél : krause@cup.hp.com
Dave Minturn, Intel Corporation ; mél : dave.b.minturn@intel.com
Jim Pinkerton, Microsoft, Inc. ; mél : jpink@microsoft.com
Hemal Shah, Broadcom Corporation ; mél : hemal@broadcom.com
Allyn Romanow, Cisco Systems ; mél : allyn@cisco.com
Tom Talpey, Network Appliance ; mél : thomas.talpey@netapp.com
Patricia Thaler, Broadcom ; mél : pthaler@broadcom.com
Jim Wendt, Hewlett Packard Corporation ; mél : jim_wendt@hp.com
Jim Williams, Emulex Corporation ; mél : jim.williams@emulex.com

Adresse des auteurs

Paul R. Culley
Hewlett-Packard Company
20555 SH 249
Houston, TX 77070-2698 USA
téléphone : 281-514-5543
mél : paul.culley@hp.com

Uri Elzur
5300 California Avenue
Irvine, CA 92617, USA
téléphone : 949.926.6432
mél : uri@broadcom.com

Renato J Recio
IBM
Internal Zip 9043
11400 Burnett Road
Austin, Texas 78759
téléphone : 512-838-3685
mél : recio@us.ibm.com

Stephen Bailey
Sandburst Corporation
600 Federal Street
Andover, MA 01810 USA
téléphone : +1 978 689 1614
mél : steph@sandburst.com

John Carrier
Cray Inc.
411 First Avenue S, Suite 600
Seattle, WA 98104-2860
téléphone : 206-701-2090
mél : carrier@cray.com

Déclaration complète de droits de reproduction