Groupe de travail Réseau
Request for Comments : 3048
Catégorie : Information
Traduction Claude Brière de L'Isle

B. Whetten, Talarian
L. Vicisano, Cisco
R. Kermode, Motorola
M. Handley, ACIRI 9
S. Floyd, ACIRI
M. Luby, Digital Fountain
janvier 2001

Blocs de construction de transport fiable de diffusion groupée pour transfert de données en vrac de un à plusieurs

Résumé
Le présent document décrit un cadre pour la normalisation du transport fiable en diffusion groupée de données en vrac. Il s'appuie sur l'expérience obtenue du déploiement de plusieurs classes de transport fiable en diffusion groupée contemporain, et tente de dégager les points communs entre ces classes de protocoles dans un certain nombre de blocs de construction. À cette fin, le présent document recommande que certains composants qui sont communs à plusieurs classes de protocoles soient normalisés comme des "blocs de construction". Les parties restantes des protocoles, consistant en fonctions très spécifiques du protocole, étroitement entrelacées, devront être désignées comme des "cœurs de protocoles". Donc, chaque protocole peut alors être construit en fusionnant un "cœur de protocole" avec un certain nombre de "blocs de construction" qui peuvent être réutilisés sur plusieurs protocoles.

Table des Matières

1. Introduction

La RFC 2357 pose les exigences pour des protocoles de diffusion groupée fiable qui sont considérés par l'IETF en vue de

leur normalisation. Cela inclut :

Contrôle d'encombrement. Le protocole doit être de déploiement sûr dans tout l'Internet. Précisément, il doit respecter trois exigences : a) il doit réaliser un bon débit (c'est-à-dire, il ne doit pas surcharger de façon constante des liaisons avec des données excessives ou du trafic de réparation) b) il doit réaliser une bonne utilisation des liaisons, et c) il ne doit pas "affamer" les flux concurrents.

Adaptabilité. Le protocole devrait être capable de fonctionner dans diverses conditions qui incluent plusieurs topologies de réseau, vitesse de liaison, et tailles d'ensembles de receveurs. Il est plus important d'avoir une bonne compréhension de comment et quand un protocole échoue que de quand et comment il fonctionne bien.

Sécurité. Le protocole doit être analysé pour montrer ce qui est nécessaire pour lui permettre de traiter les problèmes de sécurité et de confidentialité. Cela inclut de comprendre le rôle du protocole dans la confidentialité des données et l'authentification de l'envoyeur, ainsi que comment le protocole va fournir des défenses contre les attaques de déni de service.

Ces exigences sont principalement destinées à s'assurer que toutes les normes vont être sûres pour un déploiement à l'échelle de l'Internet entier. L'avancement de la maturité des travaux en cours sur le contrôle d'encombrement de la diffusion groupée fiable (RMCC, *Reliable Multicast Congestion Control*) [RFC4654] dans le groupe de recherche Diffusion groupée fiable (RMRG, *Reliable Multicast Research Group*) de l'IRTF a été un des événements qui ont permis à l'IETF de mandater le groupe de travail RMT. Le RMCC traite seulement un sous ensemble de l'espace de conception de la diffusion groupée fiable. Il se trouve fortuitement que les exigences auxquelles il répond sont aussi les plus pressantes pour les applications et le marché.

La capacité d'un protocole à satisfaire aux exigences de contrôle d'encombrement, d'adaptabilité, et de sécurité est affectée par un certain nombre d'exigences secondaires qui sont décrites dans un document séparé [RFC2887]. En résumé, ce sont :

- Garanties d'ordre. Un protocole doit offrir au moins une des garanties de livraison ordonnée ou non ordonnée de source. La prise en charge d'un ordre total sur plusieurs envoyeurs n'est pas recommandée, car cela rend plus difficile d'adapter le protocole, et peut être mis en œuvre plus facilement à plus haut niveau.

- Adaptabilité du receveur. Un protocole devrait être capable de prendre en charge un "grand" nombre de receveurs simultanés par groupe de transport. Un ensemble typique de receveurs pourrait être de l'ordre d'au moins 1 000 à 10 000 receveurs simultanés par groupe, ou pourrait même éventuellement compter des millions de receveurs dans l'Internet.

- Retours en temps réel. Certaines versions de RMCC peuvent exiger des retours en temps réel, afin qu'un protocole puisse fournir des moyens pour que ces informations soient mesurées et retournées à l'envoyeur. Bien que cela n'exige pas qu'un protocole livre les données en temps réel, c'est une importante exigence d'application qui peut être facilement fournie avec des retours en temps réel.

- Garanties de livraison. Dans de nombreuses applications, une ou des unités de données définies logiquement sont à livrer à plusieurs clients, par exemple, un fichier ou un ensemble de fichiers, un paquetage logiciel, la cote d'une action ou la cote de la bourse, une notification d'événement, un ensemble de transparents, une trame ou bloc d'une vidéo. Une unité de données d'application est définie comme étant une unité de données logiquement séparable qui est utile à l'application. Dans certains cas, une unité de données d'application peut être assez courte pour tenir dans un seul paquet (par exemple, une notification d'événement ou un cours de bourse) tandis que dans d'autres cas, une unité de données d'application peut être beaucoup plus longue qu'un paquet (par exemple, un paquetage logiciel). Un protocole doit fournir un bon débit d'unités de données d'application aux receveurs. Cela signifie que la plupart des données livrées aux receveurs sont utiles pour récupérer l'unité de données d'application qu'ils essayent de recevoir. Un protocole peut facultativement fournir une confirmation de livraison, c'est-à-dire, un mécanisme pour que les receveurs informent l'envoyeur du moment où les données sont livrées. Il y a deux types de confirmation, au niveau de l'unité de données d'application et au niveau du paquet. La confirmation d'unité de données d'application est utile au nivau de l'application, par exemple, pour informer l'application des progrès du receveur et pour décider quand arrêter d'envoyer des paquets au sujet d'une unité de données d'application particulière. La confirmation de paquet est utile au niveau du transport, par exemple, pour informer le niveau transport du moment où il peut libérer l'espace de mémoire tampon utilisé pour mémoriser les paquets pour lesquels la livraison a été confirmée. La confirmation au niveau du paquet peut aussi aider à la confirmation des unités de données d'application.

- Topologies de réseau. Un protocole ne doit pas perturber le réseau quand il est déployé dans l'Internet. Cependant, on reconnaît que les intranets vont être là où se produira la première vague de déploiements, et leur prise en charge est aussi très importante. Donc, la prise en charge des réseaux par satellite (incluant ceux qui ont des chemins de retour

terrestres ou pas de chemin de retour du tout) est encouragée, mais pas exigée.

- o Adhésion de groupe. Les algorithmes d'adhésion de groupe doivent être adaptables. L'adhésion peut être anonyme (quand l'envoyeur ne connaît pas la liste des receveurs) ou pleinement répartie (quand l'envoyeur reçoit un compte du nombre de receveurs, et facultativement, une liste des défaillances).

- o Exemple d'applications. Certaines des applications qu'un protocole de diffusion groupée fiable (RM, *Reliable Multicast*) pourrait être destiné à prendre en charge incluent des diffusions multimédia, la distribution de données de marché financier en temps réel, le transfert de fichiers en diffusion groupée, et la réplication de serveur.

Dans la suite de ce document, les termes suivants vont être utilisés avec une connotation spécifique : "famille de protocoles", "composant de protocole", "bloc de construction", "cœur de protocole", et "instanciation de protocole". Une "famille de protocoles" est une classe large de protocoles RM qui partagent une caractéristique commune. Dans notre classification, cette caractéristique est le mécanisme utilisé pour réaliser la fiabilité. Un "composant de protocole" est une partie logique du protocole qui traite une fonction particulière. Un "bloc de construction" est un constituant d'un protocole qui met en œuvre une, plus d'une, ou une partie d'un composant. Un "cœur de protocole" est l'ensemble de fonctions requises pour l'instanciation d'un protocole complet, qui n'est pas spécifié par un bloc de construction. Finalement, une "instanciation de protocole" est un protocole RM réel défini en termes de blocs de construction et de cœur de protocole.

1.1 Familles de protocoles

Le document d'espace de conception [RFC2887] donne aussi une taxonomie des approches les plus courantes qui ont été proposées au cours des dix dernières années. Après le contrôle d'encombrement, le principal défi a été de satisfaire l'exigence d'assurer un bon débit d'une façon qui s'adapte à un grand nombre de receveurs. Pour les protocoles qui incluent un canal de secours pour la récupération des paquets perdus, la capacité de tirer parti de la prise en charge des éléments du réseau a été estimée être très bénéfique pour la prise en charge d'un bon débit pour un grand nombre de receveurs. D'autres protocoles ont trouvé très bénéfique de transmettre des données codées pour réaliser un bon débit pour de grands nombres de receveurs.

Cette taxonomie divise les protocoles proposés en quatre familles. Certains protocoles de la famille fournissent la confirmation de la livraison au niveau du paquet qui peut être utile au niveau transport. Tous les protocoles dans toutes les familles peuvent être complétés par des protocoles de niveau supérieur qui fournissent une confirmation de livraison des unités de données d'application.

1. NACK seulement. Des protocoles comme SRM [FJM95] et MDP2 [MA99] tentent de limiter le trafic en utilisant seulement des NACK pour demander la retransmission de paquets. Ils n'exigent pas d'infrastructure du réseau.

2. ACK fondés sur une arborescence. Des protocoles comme RMTP [LP96], [PSLB97], RMTP-II [WBPM98] et TRAM [KCW98], utilisent des accusés de réception positifs (ACK). Les protocoles fondés sur le ACK réduisent le besoin de protocoles supplémentaires qui fournissent des confirmation de livraison, car les ACK peuvent être utilisés à cette fin. Pour éviter des explosions de ACK dans des déploiements adaptés, le protocole peut utiliser des serveurs placés dans le réseau.

3. Codage en couches asynchrone (ALC, *Asynchronous Layered Coding*). Ces protocoles (les exemples incluent [RV97] et [BLMR98]) utilisent des méthodes de correction d'erreur directe (FEC, *Forward Error Correction*) fondées sur l'envoyeur sans rétroaction de la part des receveurs ou du réseau pour assurer un bon débit. Ces protocoles utilisent aussi des protocoles de diffusion groupée mise en couche sur la base de l'envoyeur et pilotés par le receveur pour se joindre et quitter ces couches sans rétroaction à l'envoyeur pour réaliser un contrôle d'encombrement adaptable.

4. Assistance d'un routeur. Comme SRM, des protocoles tels que PGM [FLST98] et [LG97] utilisent aussi les accusés de réception négatifs pour la récupération de paquet. Ces protocoles tirent parti du nouveau logiciel de routeur pour faire des accusés de réception négatifs contraints et des retransmissions. Les protocoles à assitance de routeur peuvent aussi fournir d'autres fonctionnalités plus efficacement que les protocoles de bout en bout. Par exemple, [LVS99] montre comment l'assistance de routeur peut fournir un contrôle d'encombrement de granularité fine pour les protocoles d'ALC. Les protocoles à assistance de routeur peuvent être conçus pour compléter touts les familles de protocoles décrites ci-dessus.

Noter que la distinction en familles de protocoles n'est pas nécessairement précise et mutuellement exclusive. Les protocoles réels peuvent utiliser une combinaison des mécanismes appartenant aux différentes classes. Par exemple, des protocoles hybrides fondés sur le NACK/ACK (comme [WBPM98]) sont possibles. D'autres exemples sont les protocoles

qui appartiennent aux classes 1 à 3 qui tirent parti de la prise en charge de routeur.

2. Raison des blocs de construction

Comme spécifé dans la [RFC2357], aucun protocole de diffusion groupée fiable seul ne va probablement satisfaire les besoins de toutes les applications. Donc, l'IETF s'attend à normaliser un certain nombre de protocoles qui sont conçus pour des besoins spécifiques des applications et du réseau. Le présent document se concentre sur les exigences pour le "transfert de données en vrac de un à plusieurs", mais à l'avenir, des protocoles et blocs de construction supplémentaires sont attendus pour traiter les besoins des autres types d'applications, incluant des application de "plusieurs à plusieurs". Noter que le transfert de données en vrac ne se réfère pas à la livraison des données en temps utile, mais déclare plutôt qu'il y a une grande quantité de données à transférer dans une session. La portée et l'approche suivies pour le développement de protocoles pour ces scénarios supplémentaires va dépendre en grande partie du succès de l'approche du "bloc de construction" mise en avant dans le présent document.

2.1 Avantages des blocs de construction

Construire un grand morceau de logiciel à partir de plus petits composants modulaires est une technique bien comprise de l'ingénierie du logiciel. Certains des avantages qui peuvent en découler incluent :

- Réutilisation de spécification : des modules peuvent être utilisés dans plusieurs protocoles, ce qui réduit la quantité de temps de développement requise.

- Complexité réduite. Dans la mesure où chaque module peut être facilement défini avec une simple API, couper un grand protocole en plus petits morceaux réduit normalement la complexité totale du système.

- Temps de vérification et de débogage réduit. La complexité réduite résulte en une réduction du temps pour déboguer les modules. Il est aussi généralement plus rapide de vérifier un ensemble de plus petits modules qu'un seul plus grand module.

- Mises à jour futures plus faciles. Il y a toujours des recherches en cours sur la diffusion groupée fiable, et on s'attend à ce que l'état de l'art continue d'évoluer. Construire des protocoles avec de plus petits modules leur permet d'être plus facilement mis à niveau pour refléter les recherches futures.

- Diagnostics communs. Dans la mesure où plusieurs protocoles partagent des en-têtes de paquet communs, les analyseurs de paquet et autres outils de diagnostic peuvent être construits pour fonctionner sur plusieurs protocoles.

- Effort réduit pour les nouveaux protocoles. Lorsque de nouvelles exigences d'application introduisent le besoin de nouvelles normes, certains modules existants peuvent être réutilisés dans ces protocoles.

- Parallélisme de développement. Si les API sont définies clairement, le développement de chaque module peut se faire en parallèle.

2.2 Risques des blocs de construction

Comme la plupart des spécifications de logiciel, cette technique de couper un protocole en plus petits composants amène aussi à des compromis. Au delà d'un certain point, les inconvénients outrepassent les avantages, et il n'est pas rentable de plus subdiviser un problème. Ces risques incluent :

- Retarder le développement. Définir l'API pour la façon dont chaque module interopère prend du temps et des efforts. Lorsque le nombre de modules augmente, le nombre d'API peut augmenter à un taux plus que linéaire. Plus un composant est étroitement couplé et complexe, plus il est difficile de définir une API simple, et moins il y a d'opportunité de sa réutilisation. En particulier, le problème de comment construire et normaliser des blocs de construction de granularité fine pour un protocole de transport est difficile, et dans certains cas exige de la recherche fondamentale.

- Complexité accrue. Si il y a trop de modules, la complexité totale du système augmente en fait, à cause de la prépondérance des interfaces entre modules.

- Performances réduites. Chaque API supplémentaire ajoute un niveau de frais généraux de traitement. Si une API est insérée dans le "cas courant" du traitement de paquet, cela risque de dégrader les performances totales du protocole.

- Abandon du travail antérieur. Le développement de protocoles de transport robustes est un processus long et coûteux en temps, qui dépend lourdement des retours des déploiements réels. Une grande quantité de travail a été consacrée ces cinq dernières années aux composants de protocoles comme RMTP-II, SRM, et PGM. Tenter de réorganiser complètement ces composants risque de faire perdre le bénéfice de ce travail.

2.3 Exigences des blocs de construction

Étant donnés ces compromis, on propose qu'un bloc de construction doive satisfaire les exigences suivantes :

- Large applicabilité. Afin d'avoir l'assurance que le composant peut être réutilisé, il devrait s'appliquer sur plusieurs familles de protocoles et permettre l'évolution du composant.

- Simplicité. Afin d'avoir l'assurance que la spécification des API de composant ne va pas ralentir dramatiquement le processus de normalisation, les API doivent être simples et directes à définir. Aucune nouvelle recherche fondamentale ne devrait être faite pour définir ces API.

- Performances. Dans la mesure du possible, les blocs de construction devrait tenter d'éviter de casser le traitement de paquet de "voie rapide", ou de cas ordinaire.

3. Composants du protocole

Cette Section propose une décomposition fonctionnelle des protocoles RM de données en vrac du point de vue des composants fonctionnels fournis à une application par un protocole de transport. Elle couvre aussi certains composants qui bien qu'ils ne fassent pas nécessairement partie du protocole de transport, sont directement impactés par les exigences spécifiques du transport fiable en diffusion groupée. La section suivante spécifie les blocs de construction recommandés qui peuvent mettre en œuvre ces composants.

Bien que cette liste essaye de couvrir tous les besoins les plus courants relatifs au transport des applications de transfert de données en vrac de un à plusieurs, de nouvelles exigences d'application pourraient apparaître durant le processus de normalisation, et donc cette liste ne doit pas être interprétée comme une déclaration de ce que la couche transport devrait ou non fournir. Néanmoins, on doit souligner que certains composants fonctionnels ont été délibérément omis car ils n'ont pas paru pertinents pour le type d'application considéré (c'est-à-dire, le transfert de données en vrac de un à plusieurs). Parmi eux figurent l'ordre des messages (c'est-à-dire, ceux qui ne peuvent pas être mis en œuvre par un simple numéro de séquence) et la livraison.

Il vaut aussi de mentionner que certains des composants fonctionnels énumérés ci-dessous peuvent être exigés par d'autres composants fonctionnels et non directement par l'application (par exemple, la connaissance de la qualité de membre est généralement requise pour mettre en œuvre le fiabilité fondée sur le ACK).

La liste suivante couvre les divers composants fonctionnels de transport et les partage en sous composants.

Fiabilité des données (assurant un bon débit) |
| - Détection/notification de perte
| - Récupération de perte
| - Protection contre la perte

Contrôle d'encombrement |
| - Rétroaction sur l'encombrement
| - Régulation de taux
| - Contrôles du receveur

Sécurité

Appartenance au groupe |
| - Notification d'appartenance

| - Gestion d'appartenance

Gestion de session |
| - Suivi de l'appartenance au groupe
| - Annonce de session
| - Début/fin de session
| - Configuration/surveillance de session

Configuration d'arborescence

Noter que tous les composants ne sont pas exigés par tous les protocoles, en fonction de la définition complète du service qui est fourni par le protocole. En particulier, certains modèles de service minimaux n'exigent pas toutes ces fonctions, incluant la notification de perte, la récupération de perte, et l'appartenance au groupe.

3.1 Définition des sous composants

Détection/notification de perte. Cela inclut combien de paquets manquants sont détectés durant la transmission et comment la connaissance de ces événements est propagée à un ou plusieurs agents qui sont désignés pour récupérer des erreurs de transmission. Cette tâche soulève des problèmes majeurs d'adaptabilité et peut conduire à des explosions de rétroactions et un effondrement du débit si elle n'est pas traitée de façon appropriée. Les mécanismes fondés sur les accusés de réception fondés sur l'arborescence (TRACK, *TRee-based positive Acknowledgement*) ou les accusés de réception négatifs (NACK) sont les plus largement utilisés pour effectuer cette fonction. Des mécanismes fondés sur une combinaison de TRACK et de NACK sont aussi possibles.

Récupération de perte. Cette fonction répond aux événements de notification de perte par la transmission de paquets supplémentaires, soit sous forme de copies de ces paquets perdus, soit sous forme de paquets de FEC. La manière de mettre en œuvre cette fonction peut significativement affecter l'adaptabilité d'un protocole.

Protection contre la perte. Cette fonction tente de masquer les pertes de paquet afin qu'elles ne deviennent pas des événements de notification de perte. Cette fonction peut être réalisée par la transmission proactive de paquets de FEC. Chaque paquet de FEC est créé à partir d'une unité de données d'application entière [LMSSS97] ou d'une portion d'une unité de données d'application [RV97], [BKKKLZ95], un fait qui permet à un receveur de récupérer de certaine perte de paquet sans autres retransmissions. Le nombre de pertes qui peuvent être récupérées sans exiger de retransmission dépend de la quantité de paquets de FEC envoyés préalablement. La protection contre la perte peut aussi être poussée à l'extrême quand un bon débit est réalisé sans aucune fonction de détection/notification de perte et de récupération de perte, comme dans la famille de protocoles d'ALC définie précédemment.

Rétroactions d'encombrement. Pour les protocoles de contrôle d'encombrement pilotés par l'envoyeur, le receveur doit fournir un type de rétroaction sur l'encombrement à l'envoyeur. Cela implique normalement des mesures de taux de pertes et de délai d'aller-retour.

Régulation de taux. Connaissant les retours d'encombrement, l'envoyeur doit alors ajuster son taux d'une façon équitable pour le réseau. Une proposition qui définit cette notion d'équité et les autres exigences de contrôle d'encombrement est [Whetten99].

Contrôles du receveur. Afin d'éviter de permettre à un receveur qui a une connexion extrêmement lente avec l'envoyeur d'arrêter toute progression dans des schémas à un seul taux, un algorithme de contrôle d'encombrement va souvent exiger que des receveurs quittent les groupes. Pour des approches à plusieurs taux, les receveurs de toutes les vitesses de connexion peuvent avoir des livraisons de données en accord avec le débit de leur connexion sans ralentir les autres receveurs.

Sécurité. La sécurité pour la diffusion groupée fiable contient un certain nombre de problèmes complexes et délicats qui s'étendent sur une grande partie du modèle de service de diffusion groupée IP. Dans ce modèle de service, les hôtes n'envoient pas de trafic à un autre hôte, mais choisissent plutôt de recevoir du trafic provenant d'un groupe de diffusion groupée. Cela signifie que tout hôte peut se joindre à un groupe et recevoir son trafic. À l'inverse, les hôtes peuvent aussi quitter le groupe à tout moment. Donc, le protocole doit traiter de la façon dont il impacte les questions de sécurité suivantes :

- Authentification de l'envoyeur (car tout hôte peut envoyeur à un groupe),
- Chiffrement des données (car tout hôte peut se joindre à un groupe),
- Protection du transport (attaques de déni de service, par la corruption de l'état de transport, ou de demandes pour des

ressources non autorisées),
- o Gestion de clé de groupe (car les hôtes peuvent se joindre et quitter un groupe à tout moment) [RFC2627].

En particulier, un protocole de transport doit porter une attention particulière à la façon dont il se protège contre les attaques de déni de service, par des mécanismes comme une authentification légère des paquets de contrôle [HW99].

Avec le modèle de service de diffusion groupée spécifique de la source (SSM, *Source Specific Multicast*) un hôte se joint spécifiquement à une paire d'envoyeur et groupe. Donc, le SSM offre plus de sécurité contre la réception par les hôtes de trafic provenant d'une attaque de déni de service où un envoyeur arbitraire envoie des paquets que les hôtes n'ont pas précisément demandé de recevoir. Néanmoins, il est recommandé que des protections supplémentaires contre de telles attaques soient fournies quand on utilise SSM, parce que la protection offerte par SSM contre de telles attaques peut n'être pas suffisante.

Authentification de l'envoyeur, chiffrement des données, et gestion de clé de groupe. Bien que ces fonctions ne fassent normalement pas partie de la couche transport par elles-mêmes, un protocole a besoin de comprendre quelles ramifications elles ont sur la sécurité des données, et peut avoir besoin d'interfaces spéciales pour la couche de sécurité afin de s'accommoder de ces ramifications.

Protection du transport. La principale tâche de sécurité pour une couche de transport est de protéger la couche de transport elle même contre les attaques. La plus importante fonction pour cela est normalement une authentification légère des paquets de contrôle afin d'empêcher la corruption de l'état et autres attaques de déni de service.

Notification d'adhésion. C'est la fonction par laquelle la source des données -- ou un agent de niveau supérieur dans une organisation éventuellement hiérarchique -- apprend l'identité et/ou le nombre des receveurs ou d'agents de niveau inférieur. Pour être adaptable, cela ne va normalement pas fournir une connaissance totale de l'identité de chaque receveur.

Gestion des adhésions. Cela met en œuvre les mécanismes pour que les membres se joignent et quittent le groupe, pour accepter/refuser de nouveaux membres, ou pour terminer l'adhésion des membres existants.

Suivi des membres du groupe. Comme caractéristique facultative, un protocole peut assurer l'interface avec un composant qui suit l'identité de chaque receveur dans un grand groupe. Si il en est ainsi, cette caractéristique va normalement être mise en œuvre hors bande, et peut être mise en œuvre par un protocole de niveau supérieur. Cela peut être utile pour des services qui exigent le suivi de l'usage du système, la facturation, et les rapports d'utilisation.

Annonce de session. Cela publie le nom/contenu de session et les paramètres nécessaires pour sa réception. Cette fonction est généralement effectuée par un protocole de couche supérieure (par exemple, [RFC2974] et [RFC2327]).

Début/arrêt de session. Ces fonctions déterminent le moment de début/arrêt de l'envoyeur et/ou des receveurs. Dans de nombreux cas, ceci est implicite ou effectué par une application ou protocole de niveau supérieur. Dans certains protocoles cependant, c'est une tâche qui est mieux effectuée par la couche de transport à cause des exigences d'adaptabilité.

Configuration/surveillance de session. Du fait de la portée potentiellement grande d'une session de diffusion groupée, il est particulièrement important qu'un protocole inclue des outils pour configurer et surveiller le fonctionnement du protocole.

Configuration d'arborescence. Pour les protocoles qui incluent des éléments hiérarchiques (comme PGM et RMTP-II) il est important de configurer ces éléments d'une façon approximativement congruente avec la topologie d'acheminement de diffusion groupée. Bien que la configuration d'arborescence pourrait être incluse au titre des outils de configuration de session, il est clairement préférable que cette configuration puisse être rendue automatique.

4. Recommandations sur les blocs de construction

Les familles de protocoles introduites au paragraphe 1.1 utilisent généralement des mécanismes différents pour mettre en œuvre les composants fonctionnels de protocole décrits à la Section 3. Cette Section essaye de grouper ces mécanismes en macro composants qui définissent des blocs de construction de protocole.

Un bloc de construction est défini comme un "composant logique de protocole qui résulte en API explicites à utiliser par

d'autres blocs de construction ou par le client de protocole."

Les blocs de construction sont généralement spécifiés en termes d'ensemble d'algorithmes et formats de paquet qui mettent en œuvre les composants fonctionnels de protocole. Un bloc de construction peut aussi avoir des API à travers lesquelles il communique avec les applications et/ou les autres blocs de construction. La plupart des blocs de construction devraient aussi avoir une API de gestion, par laquelle ils communiquent avec SNMP et/ou autres protocoles de gestion.

Dans les paragraphes qui suivent, on va faire une liste d'un certain nombre de blocs de construction qui, à ce stade, semblent couvrir la plupart des composants fonctionnels nécessaires pour mettre en œuvre les familles de protocoles présentées au paragraphe 1.1. Néanmoins, cette liste représente la "meilleure hypothèse courante", et à ce titre n'est pas conçue comme étant exhaustive. La décomposition actuelle en blocs de construction, c'est-à-dire, la division des composants fonctionnels en blocs de construction, peut aussi devoir être révisée à l'avenir.

4.1 Fiabilité fondée sur le NACK

Ce bloc de construction définit la détection/notification et la récupération de pertes fondée sur le NACK. Les problèmes majeurs qu'il vise sont la prévention (suppression) d'explosions et la sémantique de NACK (c'est-à-dire, comment les paquets à retransmettre devraient être spécifiés, dans les deux cas de réparation de perte sélective et de FEC). Les mécanismes de suppression à considérer sont :

- o les NACK de diffusion groupée,
- o les NACK en envoi individuel et confirmation en diffusion groupée.

Ces mécanismes de suppression ont principalement besoin de minimiser le délai tout en minimisant aussi les messages redondants. Ils peuvent aussi avoir besoin d'une pondération spéciale pour fonctionner avec les rétroactions d'encombrement.

4.2 Codage de FEC

Ce bloc de construction est concerné par les informations de FEC de niveau paquet quand les codes de FEC sont utilisés proactivement ou comme réparation en réaction à la perte de paquets. Il spécifie le choix de code de FEC et le nom de paquet de FEC (indexation) pour la réparation de FEC réactive et la FEC pro-active.

4.3 Contrôle d'encombrement

Il va probablement y avoir plusieurs versions de ce bloc de construction, correspondant aux différentes politiques de conception pour traiter le contrôle d'encombrement. Deux approches principales sont considérées pour l'instant : une régulation de taux fondées sur la source avec un seul taux fourni à tous les receveurs dans la session, et une approche à plusieurs taux pilotée par le receveur avec différents receveurs recevant à des taux différents dans la même session. L'approche à plusieurs taux peut utiliser plusieurs couches de trafic de diffusion groupée [VRC98] ou un filtrage de routeur sur une seule couche [LVS99]. L'approche à plusieurs taux est la plus applicable pour les protocoles d'ALC.

Les deux approches sont encore en phase d'étude, cependant la première semble être assez mûre [RFC4654] pour permettre de commencer le processus de normalisation.

Au moment de la rédaction du présent document, une troisième classe d'algorithmes de contrôle d'encombrement fondée sur la prise en charge du routeur commence à émerger dans le RMRG de l'IRTF [LVS99]. Ce travail peut conduire à la future normalisation d'un ou plusieurs blocs de construction supplémentaires pour le contrôle d'encombrement.

4.4 Prise en charge de routeur générique

La tâche de concevoir des protocoles RM peut être rendue plus aisée en présence d'une prise en charge spécifique dans les routeurs. Dans certains cas spécifiques d'application, les avantages accrus de l'ajout d'une prise en charge supplémentaire de routeur peuvent justifier la complexité et les dépenses supplémentaires résultantes [FLST98].

Les composants fonctionnels qui peuvent tirer parti de la prise en charge de routeur incluent de l'agrégation/suppression de rétroactions (à la fois pour la notification de pertes et le contrôle d'encombrement) et la retransmission contrainte des paquets de réparation. Un autre composant qui peut tirer parti de la prise en charge de routeur support est le filtrage intentionnel de paquets pour fournir des taux de livraison des paquets différents aux différents receveurs provenant du

même flux de paquets en diffusion groupée. Cela pourrait être des plus avantageux quand il est combiné avec des protocoles d'ALC [LVS99].

Le processus de conception et de déploiement de ces mécanismes à l'intérieur des routeurs peut être beaucoup plus lent que celui exigé pour les mécanismes de protocole d'hôte d'extrémité. Donc, il serait très avantageux de définir ces mécanismes d'une façon générique que plusieurs protocoles peuvent utiliser si il est disponible, mais n'a pas nécessairement besoin d'en dépendre.

Ce composant a deux moitiés, un protocole de signalisation et des algorithmes réels de routeur. Le protocole de signalisation permet au protocole de transport de demander au routeur les fonctions qu'il souhaite effectuer, et les algorithmes de routeur d'effectuer réellement ces fonctions. Il est plus urgent de définir le protocole de signalisation, car il va probablement impacter les en-têtes de protocole courants.

Un composant important du protocole de signalisation est un certain niveau de correspondance entre les en-têtes de paquet de plusieurs protocoles, qui permet au routeur de reconnaître et interpréter les en-têtes.

4.5 Configuration d'arborescence

Il a été montré que l'adaptabilité des protocoles RM peut être largement améliorée par l'insertion d'une forme de d'agents d'agrégation de retransmission ou de rétroaction entre la source et les receveurs. Ces agents sont alors utilisés pour former une arborescence avec la source à (ou proche de) la racine, les receveurs aux feuilles de l'arborescence, et les nœuds de réparation d'agrégation/locaux au milieu. Les nœuds internes peuvent soit être un logiciel dédié pour cette tâche, soit des receveurs qui effectuent cette double tâche.

L'efficacité de ces agents pour aider à la livraison des données est étroitement dépendante de la façon dont l'arborescence logique qu'ils utilisent pour communiquer correspond à la topologie d'acheminement sous-jacente. L'objet de ce bloc de construction va être de construire et gérer l'arborescence logique qui connecte les agents. Idéalement, ce bloc de construction va effectuer ces fonctions d'une manière qui s'adapte aux changements des membres de la session, de la topologie de l'acheminement, et de la disponibilité du réseau.

4.6 Sécurité des données

Au moment de la rédaction du présent mémoire, les problèmes de sécurité sont l'objet de recherches au sein du groupe Diffusion groupée sûre (SMuG, *Secure Multicast Group*) de l'IRTF. Les solutions pour ces exigences seront normalisées au sein de l'IETF quand elles seront prêtes.

4.7 En-têtes communs

Comme mentionné dans la section sur la prise en charge générique de routeur, il est important d'avoir un certain niveau de correspondance dans les en-têtes de paquet. Il peut aussi être utile d'avoir des formats d'en-tête de données commun s pour d'autres raisons. Ce bloc de construction consisterait en recommandations sur les champs dans leurs en-têtes de paquet que les protocoles devraient rendre communs entre eux.

4.8. Cœurs de protocoles

Les blocs de construction ci-dessus consistent en les composants fonctionnels énumérés à la Section 3 qui paraissent satisfaire les exigences pour être mis en œuvre comme les blocs de construction présentés à la Section 2.

Les autres fonctions de la Section 3, qui ne sont pas couvertes ci-dessus, devraient être mises en œuvre au titre des "cœurs de protocoles", spécifiques de chaque protocole normalisé.

5. Considérations sur la sécurité

La RFC 2357 déclare spécifiquement que "les projets Internet sur la diffusion groupée fiable revus par les directeurs de la zone Transport doivent explicitement explorer les aspects de sécurité de la conceptin proposée". Spécifiquement, le travail en cours sur le bloc de construction RMT doit examiner les attaques de déni de service qui peuvent être faites contre les blocs de construction et affectés par les blocs de construction sur l'Internet au sens large. Cette exigence s'ajoute aux

discussions concernant la sécurite des données, qui est la manipulation ou l'exposition des informations de session à des receveurs non autorisés. Les lecteurs se reporteront au paragraphe 5.e de la RFC 2357 pour les détails.

6. Considérations relatives à l'IANA

Il va y avoir plus d'un bloc de construction, et éventuellement plusieurs versions de blocs de construction individuels lorsque leur conception sera précisée. Pour cette raison, la création de nouveaux blocs de construction et de nouvelles versions de bloc de construction sera administrée via un registre des blocs de construction qui va être administré par l'IANA. Initialement, ce registre sera vide, car les blocs de construction décrits dans les paragraphes 4.1 à 4.3 sont présentés à des fins d'exemple et de concept. Le registre des blocs de construction demandé à l'IANA sera rempli à partir des spécifications au fur et à mesure de leur approbation comme RFC publiées (en utilisant la politique de "Spécification exigée" comme décrit dans la [RFC2434]). Un enregistrement va consister en un nom de bloc de construction, un numéro de version, un bref texte descriptif, un numéro de RFC, et une personne responsable, auquel l'IANA va allouer le numéro de type.

7. Conclusions

Dans ce document, on a décrit brièvement un certain nombre de blocs de construction qui peuvent être utilisés pour générer des protocoles de diffusion groupée fiable à utiliser dans l'espace d'application de transfert de données en vrac de un à plusieurs. La liste des blocs de construction présentée a été déduite de la considération des fonctions qu'un protocole dans cet espace doit effectuer et de comment ces fonctions devraient être groupées. Cette liste n'est pas destinée à tout inclure mais plutôt d'être un guide sur la façon dont les blocs de construction sont considérés durant le processus de normalisation au sein du groupe de travail Transport de diffusion groupée fiable.

8. Remerciements

Le présent document représente un survol d'un certain nombre de blocs de construction pour le transfert de données en vrac de un à plusieurs qui peuvent être prêts pour la normalisation au sein du groupe de travail RMT. Les idées présentées ne sont pas celles des auteurs, mais sont plutôt le résultat de l'accumulation sur de nombreuses années des recherches dans le transport de diffusion groupée combinée avec les diverses présentations et discussions au sein du groupe de recherche Diffusion groupée fiable de l'IRTF. Bien qu'ils soient trop nombreux pour qu'on les mentionnent tous ici, nous remercions chacun de ceux qui ont participé à ces discussions pour leurs contributions.

9. Références

[BKKKLZ95] J. Bloemer, M. Kalfane, M. Karpinski, R. Karp, M. Luby, D. Zuckerman, "An XOR-based Erasure Resilient Coding Scheme", Rapport technique ICSI n° TR-95-048, août 1995.

[BLMR98] J. Byers, M. Luby, M. Mitzenmacher, A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data", Proc ACM SIGCOMM 98.

[FJM95] S. Floyd, V. Jacobson, S. McCanne, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", Proc ACM SIGCOMM 95, août 1995 pp. 342-356.

[FLST98] D. Farinacci, S. Lin, T. Speakman, and A. Tweedly, "PGM reliable transport protocol specification", Travail en cours.

[HW99] T. Hardjorno, B. Whetten, "Security Requirements for RMTP-II", Travail en cours, jiun 1999.

[KCW98] M. Kadansky, D. Chiu, and J. Wesley, "Tree-based reliable multicast (TRAM)", Travail en cours.

[Kermode98] R. Kermode, "Scoped Hybrid Automatic Repeat Request with Forward Error Correction", Proc ACM SIGCOMM 98, septembre 1998.

[LDW98] M. Lucas, B. Dempsey, A. Weaver, "MESH: Distributed Error Recovery for Multimedia Streams in Wide-Area Multicast Networks".

[LESZ97] C-G. Liu, D. Estrin, S. Shenkar, L. Zhang, "Local Error Recovery in SRM: Comparison of Two Approaches", USC Technical Report 97-648, janvier 1997.

[LG97] B.N. Levine, J.J. Garcua-Luna-Aceves, "Improving Internet Multicast Routing with Routing Labels", IEEE International Conference on Network Protocols (ICNP-97), 28-31 octobre 1997, p. 241-250.

[LP96] K. Lin and S. Paul. "RMTP: A Reliable Multicast Transport Protocol", IEEE INFOCOMM 1996, mars 1996, pp. 1414-1424.

[LMSSS97] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, V. Stemann, "Practical Loss-Resilient Codes", Proc ACM Symposium on Theory of Computing, 1997.

[LVS99] M. Luby, L. Vicisano, T. Speakman. "Heterogeneous multicast congestion control based on router packet filtering", RMT working group, juin 1999, Pisa, Italy.

[MA99] J. Macker, B. Adamson. "Multicast Dissemination Protocol version 2 (MDPv2)", Travail en cours, http://manimac.itd.nrl.navy.mil/MDP

[OXB99] O. Ozkasap, Z. Xiao, K. Birman, "Scalability of Two Reliable Multicast Protocols", Travail en cours, mai 1999.

[PSLB97] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)", IEEE Journal on Selected Areas in Communications, Vol. 15, n° 3, avril 1997.

[RFC2327] M. Handley et V. Jacobson, "SDP : Protocole de description de session", avril 1998. *(Obsolète; voir* RFC4566)

[RFC2357] A. Mankin, A. Romanov, S. Bradner et V. Paxson, "Critères de l'IETF pour l'évaluation des protocoles de transport et d'application de diffusion groupée fiable", juin 1998. *(Information)*

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Rendue obsolète par la* RFC5226)

[RFC2627] D. Wallner, E. Harder, R. Agee,"Gestion de clés en diffusion groupée : problèmes et architectures", juin 1999. *(Info.)*

[RFC2887] M. Handley et autres, "Espace de conception de diffusion groupée fiable pour transfert de données brutes", août 2000. *(Info.)*

[RFC2974] M. Handley, C. Perkins, E. Whelan, "Protocole d'annonce de session (SAP)", octobre 2000. *(Expérimentale)*

[RFC4654] J. Widmer, M. Handley, "Spécification du protocole de contrôle d'encombrement de diffusion groupée compatible TCP (TFMCC)", août 2006. *(Expérimentale)*

[RV97] L. Rizzo, L. Vicisano, "A Reliable Multicast Data Distribution Protocol Based on Software FEC Techniques", Proc. of The Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS'97), Sani Beach, Chalkidiki, Greece, 23-25 juin 1997.

[VRC98] L. Vicisano, L. Rizzo, J. Crowcroft, "TCP-Like Congestion Control for Layered Multicast Data Transfer", Proc. of IEEE Infocom'98, mars 1998.

[WBPM98] B. Whetten, M. Basavaiah, S. Paul, T. Montgomery, N. Rastogi, J. Conlan, and T. Yeh, "THE RMTP-II PROTOCOL", Travail en cours.

[Whetten99] B. Whetten, "A Proposal for Reliable Multicast Congestion Control Requirements", Travail en cours. http://www.talarian.com/rmtp-ii/overview.htm

10. Adresse des auteurs

Michael Luby
Digital Fountain
600 Alabama Street
San Francisco, CA 94110
mél : luby@digitalfountain.com

Lorenzo Vicisano
Cisco Systems, Inc.
170 West Tasman Dr.,
San Jose, CA, USA, 95134
mél : lorenzo@cisco.com

Brian Whetten
Talarian Corporation,
333 Distel Circle,
Los Altos, CA 94022, USA
mél : whetten@talarian.com

Roger Kermode
Motorola Australian Research Centre
Level 3, 12 Lord St,
Botany NSW 2019, Australia
mél : Roger.Kermode@motorola.com

Mark Handley
ICSI Center for Internet Research
1947 Center St., Suite 600,
Berkeley CA, 94704 USA
mél : mjh@icir.org

Sally Floyd
ATT Center for Internet Research at ICSI,
International Computer Science Institute,
1947 Center Street
Berkeley, CA 94704, USA
mél : floyd@aciri.org

11. Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'éditeur des RFC est actuellement fourni par la Internet Society.