Influence du temps d'adhésion sur le contrôle de congestion multicast

Vincent Lucas, Jean-Jacques Pansiot, Mickaël Hoerdt

Influence du temps d'adhésion sur le contrôle de congestion multicast

Vincent Lucas* — **Jean-Jacques Pansiot**** — **Mickaël Hoerdt*****

Laboratoire LSIIT (UMR 7005) - Université Louis Pasteur
Boulevard Sébastien Brant - BP 10413
F-67412 Illkirch Cedex

lucas@clarinet.u-strasbg.fr, pansiot@crc.u-strasbg.fr***

Norvegian University of Science and Technology - NTNU
E-261 - Elektro E - Q2S Center of excellence
NO-7491 Trondheim - Norway

*hoerdt@q2s.ntnu.no****

RÉSUMÉ. *Bien que le routage IP multicast soit maintenant bien défini, certaines lacunes ralentissent son déploiement, notamment au niveau du contrôle de congestion qui est indispensable au bon fonctionnement d'Internet. De RLM à FLID-SL, les premiers mécanismes de contrôle de congestion souffrent de la latence du temps de désabonnement. Pour résoudre cela, FLID-DL et WEBRC utilisent des méthodes à canaux dynamiques, mais reposent sur l'hypothèse que le temps d'adhésion est faible. Cet article présente une étude du temps d'adhésion et montre que cette hypothèse n'est pas toujours vérifiée. Cette constatation nous pousse à analyser l'influence du temps d'adhésion sur les mécanismes de contrôle de congestion. Nous avons réalisé des tests sur une maquette locale et sur le m6bone qui illustrent ce problème. Enfin, nous proposons pour solution un mécanisme utilisant la variation du temps de propagation; solution validée par des tests, dont nous analysons les premiers résultats.*

ABSTRACT. *Although multicast IP routing is well defined, some missing mechanisms, such as congestion control, slow its deployment. From RLM to FLID-SL, the first congestion control schemes suffer from leave delay latency. In order to solve this problem, FLID-DL and WEBRC use methods based on dynamic channels. However, these proposals are based on the assumption that join time is small. This article presents a study of join time and shows that this assumption is not always verified. This observation leads us to analyse the influence of join time on the different congestion control methods. Tests realized on a local network and on the m6bone illustrate this problem. Finally, we propose a solution using the variation of the propagation delay. This solution is under test, and we analyse the first results.*

MOTS-CLÉS : *Multicast, contrôle de congestion, temps d'adhésion, WEBRC, FLID-DL, FLID-SL*

KEYWORDS: *Multicast, congestion control, join time, WEBRC, FLID-DL, FLID-SL*

1. Introduction

Depuis quelques années, le routage IP multicast, notamment PIM (Protocol Independent Multicast), est implémenté dans la plupart des routeurs. Malgré cela, il est souvent désactivé sur les routeurs des différents opérateurs. Ce comportement provient du constat qu'il manque encore certains moyens de gestion du multicast, principalement pour des raisons de facturation et de difficultés de gestion des flux. De plus, le multicast ne dispose pas actuellement d'un mécanisme de diffusion fiable et performant. Une telle diffusion se décompose en deux parties qui sont un contrôle de congestion et une correction des pertes. Pour la correction des pertes, les choix se sont portés sur l'utilisation des codes FEC [LUB 05] [LUB 02b] (Forward Error Correction). Ce type de code ajoute des paquets supplémentaires à la diffusion, prodiguant ainsi un pouvoir correcteur. Ce mécanisme permet de récupérer les données initiales, sans avoir ni à acquitter, ni à demander la retransmission des paquets perdus. Bien entendu, cette méthode ne corrige qu'un certain pourcentage de pertes.
Depuis la création de RLM (Receiver-driven Layered Multicast), les mécanismes de contrôle de congestion extensibles pour le multicast ne cessent d'évoluer, formant maintenant une famille comprenant notamment : FLID-SL [LUB 00] (Fair Layered Increase/Decrease with Static Layering), FLID-DL [BYE 02] (Fair Layered Increase/Decrease with Dynamic Layering), WEBRC [LUB 02a] [LUB 04] (Wave and Equation Base Rate Control). Tous ces protocoles se basent sur l'utilisation de plusieurs canaux organisés de manière hiérarchique. Chaque récepteur peut ainsi ajuster son débit en s'abonnant au nombre de canaux désirés pour atteindre un débit optimal. Ces protocoles prennent en compte le fait que le temps de désabonnement d'un canal multicast est particulièrement long car ce temps peut atteindre 3 secondes en IGMPv3/MLDv2. Un autre point commun à ces méthodes est qu'elles se basent sur l'hypothèse implicite que le temps d'adhésion est négligeable et n'est fonction que du temps de propagation. Cependant, certains chercheurs et utilisateurs du m6bone [1] ont remarqué que le comportement réel du temps d'adhésion diffère de cette logique. Ces observations ont motivé la rédaction de cet article, qui en premier lieu, porte sur une étude du temps d'adhésion pour montrer et analyser les erreurs de cette hypothèse. La seconde partie sera l'étude des répercussions que le comportement du temps d'adhésion peut avoir sur les méthodes de contrôle de congestion multicast.

Cet article est organisé de la façon suivante. La section 2 présente les différents mécanismes de contrôle de congestion. La section 3 définit ce qu'est le temps d'adhésion. La section 4 présente une étude des temps d'adhésion relevés. La section 5 contient une série de tests révélant l'influence du temps d'adhésion sur les mécanismes de contrôle de congestion. La section 6 propose une solution préliminaire pour faire face aux problèmes détectés. La section 7 présente une méthode permettant d'améliorer la réactivité côté récepteur. Enfin la section 8 synthétise et conclut cette étude.

1. Le m6bone est une part d'Internet IPv6, le 6bone, où le multicast est activé.

2. Présentation des mécanismes de contrôle de congestion

Tous les mécanismes de contrôle de congestion que nous allons présenter utilisent des canaux hiérarchiques. Chaque mécanisme utilise un ensemble de canaux qui forment une session où chaque canal correspond à un groupe multicast. Un récepteur doit donc s'abonner à plus ou moins de canaux simultanément pour régler son débit. Par exemple pour un flux vidéo ajouter un canal augmente la qualité de la vidéo reçue. Dans le cas de transmission fiable, comme avec FLUTE [PAI 04], chaque canal supplémentaire augmente le débit du transfert. Chaque paquet est estampillé par un numéro d'intervalle noté TSI (Time Slot Interval) incrémenté toutes les TSD (Time Slot Duration) secondes. Les mécanismes de contrôle de congestion se divisent en deux grandes familles qui utilisent des canaux statiques comme FLID-SL, ou des canaux dynamiques comme FLID-DL et WEBRC.

– Une source utilisant des canaux statiques émet en continu sur chacun de ses canaux. Le récepteur doit donc s'abonner ou se désabonner pour respectivement augmenter ou réduire son débit. Pour FLID-SL, chaque changement de TSI marque une décision du récepteur. Il décide d'augmenter son débit, si, pendant le TSI précédent il n'a pas connu de pertes et qu'un marqueur était activé sur son canal le plus haut. S'il a connu des pertes, le récepteur doit alors réduire son débit. Sinon son débit ne change pas. Ainsi l'augmentation ou la réduction effective du débit est dépendante de la rapidité d'abonnement ou de désabonnement des protocoles de routage multicast. Cela pose un problème de latence au désabonnement car il faut déjà 3 secondes pour se désabonner au niveau local avec IGMPv3/MLDv2.

– L'utilisation de canaux dynamiques permet de s'affranchir des lenteurs de désabonnement d'un canal. Pour cela, la source émet de façon constante uniquement sur le canal de base. Les autres canaux diminuent régulièrement leur débit d'un facteur donné. Pour FLID-DL la diminution de débit se fait au changement de TSI, alors que pour WEBRC cette diminution est continue dans le temps. A chaque changement de TSI, le canal dynamique le plus bas devient alors muet pour une longue période et un nouveau canal est créé avec un débit maximal. Ainsi un récepteur doit toujours se désabonner du canal qui vient de devenir muet. S'il ne fait rien d'autre, son débit diminue. S'il veut conserver son débit, il doit s'abonner à un nouveau canal. Et s'il veut augmenter son débit, il doit s'abonner à un second canal avant la fin du nouveau TSI. FLID-DL prend ses décisions exactement de la même manière que FLID-SL. Pour WEBRC, chaque décision se fait à un changement d'époque qui arrive 20 fois par TSI en comparant une estimation de la bande passante disponible notée $reqn$ et une prévision du débit attendu pour la prochaine époque, notée arr_p. Le calcul de $reqn$ se fait suivant l'équation 1 qui a été proposée dans [LUB 04] pour approcher la formule de Padhye [PAD 98] modélisant le débit TCP.

$$reqn = \frac{1}{artt * \sqrt{lossp} * (0.816 + 7.35 * lossp * (1 + 32 * lossp^2))} \quad [1]$$

$reqn$ repose sur une probabilité de pertes notée $lossp$, et sur une approximation du RTT multicast notée $artt$. Quant à arr_p, il repose sur le débit idéalement reçu à

l'époque précédente réduit du facteur de diminution des canaux. Par conséquent, si $arr\_p \leq reqn$, alors le récepteur décide d'augmenter sa réception. Sinon le débit du récepteur diminue tout seul tant que $arr\_p > reqn$.

Des mécanismes basés sur l'envoi de paires de paquets (packet pair probes) ont été proposés pour évaluer la congestion comme PLM [LEG 00] ou ERA [PUA 03]. Néanmoins PLM nécessiterait l'implantation de « Fair Queuing » dans les routeurs, et ERA ne propose pas de méthode pour estimer le RTT et nous ne l'étudierons pas plus avant. [BYE 06] propose aussi une autre approche utilisant des canaux non hiérarchiques.

3. Nature et définition du temps d'adhésion

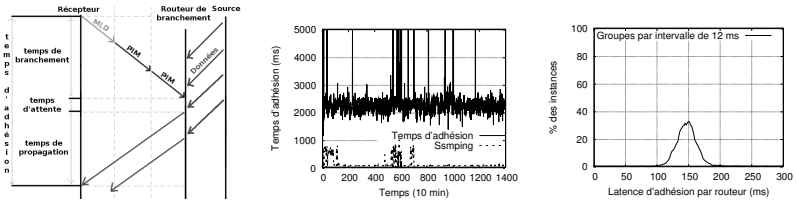
La littérature [QUI 01] définit le temps d'adhésion comme étant le délai séparant le moment où un récepteur envoie un « IGMP/MLD report » pour s'abonner à un groupe multicast et le moment où ce même récepteur reçoit le premier paquet du groupe demandé. Ce temps se compose de trois phases (figure 1(a)) :

– **Un temps de branchement** : Il correspond à l'intervalle de temps qui s'écoule entre la demande d'abonnement à un groupe multicast et le moment où la nouvelle branche créée par cette demande est connectée à l'arbre existant du même groupe. Ce temps se découpe lui-même en un temps de signalisation entre récepteur et routeur (Report IGMP ou MLD) et un temps de signalisation entre routeurs (PIM join). Pour la suite, nous utiliserons le terme « branchement » pour désigner le fait de réaliser une connexion entre la nouvelle branche et l'arbre existant. De même, nous nommerons « routeur de branchement » le routeur au sein duquel se déroule le branchement.

– **Un temps d'attente** : Il correspond au temps séparant la réalisation du branchement et l'arrivée du premier paquet du canal au niveau du routeur de branchement.

– **Un temps de propagation** : Il correspond à l'intervalle de temps entre l'arrivée du premier paquet du canal sur le routeur de branchement et la réception du même paquet par le récepteur.

Cette décomposition nous permet de définir les éléments qui peuvent faire varier le temps d'adhésion. La signalisation devant se propager saut par saut jusqu'au routeur de branchement, plus le routeur de branchement est éloigné du récepteur, plus le temps d'adhésion sera important. En second point, on peut remarquer que la probabilité que le routeur de branchement soit proche varie en fonction du nombre de récepteurs actuellement abonnés à la même session. En effet, plus le nombre de récepteurs est élevé, plus l'arbre multicast existant a de chances de recouvrir une grande partie du chemin entre la source et le récepteur, et plus le routeur de branchement est proche du récepteur. Troisièmement, les mesures du temps d'adhésion présentent des différences de résultats en fonction de la fréquence des envois des paquets par la source, qui influe sur le temps d'attente. Dans la suite de l'étude, il faudra donc garder à l'esprit que les mesures des temps d'adhésion sont beaucoup plus précises quand le débit de la source est important. Enfin le temps d'adhésion varie en fonction du traitement des messages d'adhésion dans les routeurs, point que nous détaillons dans la partie suivante.



(a) Description (b) Mesures (c) Répartition

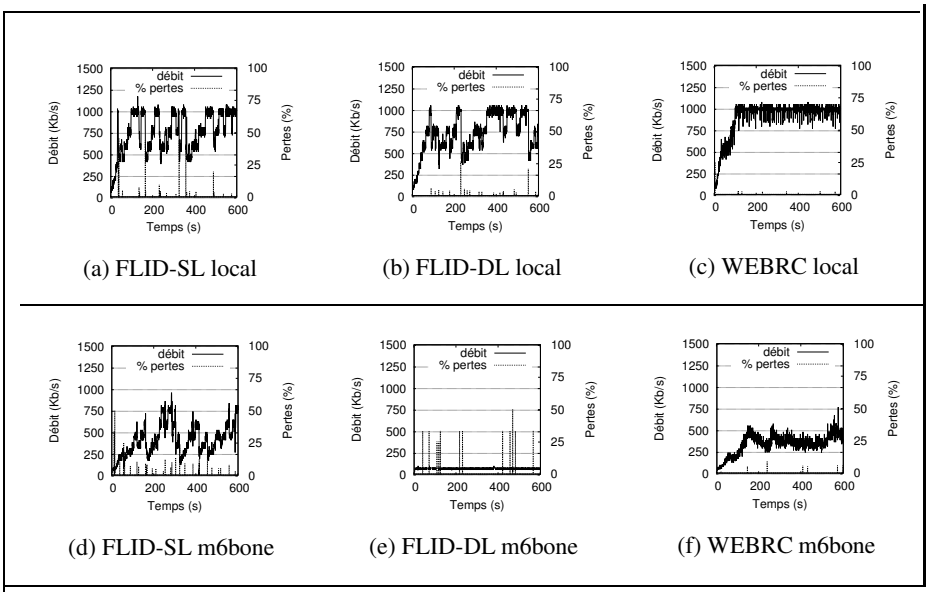
Figure 1. *Description du temps d'adhésion et résultats des tests sur le m6bone*

4. Étude des temps d'adhésion

Comme nous l'avons présenté dans la section précédente, le temps d'adhésion est sensible à plusieurs paramètres. Le traitement des messages d'adhésion au sein des routeurs ne semble pas être effectué de la même manière suivant l'implémentation utilisée. Plusieurs observations concordent, affirmant que contrairement aux implémentations telles que « MRD6 » [2], les routeurs des grands équipementiers semblent construire relativement lentement l'arbre multicast. Pour étudier ce comportement, nous avons effectué des tests sur le m6bone entre Strasbourg et Trondheim, avec 15 sauts multicast entre les machines de tests. Les tests se sont déroulés en multicast SSM-IPv6, avec des relevés effectués toutes les 10 minutes par l'outil « multicat » pour le temps d'adhésion et « ssmping » [3] pour le temps de propagation.
Les résultats des ces tests sont présentés par la figure 1(b). Le temps de propagation est de 50 à 60 ms en temps normal, mais peut atteindre 1000 ms en cas de forte congestion. Le temps d'adhésion reste toujours autour de 2.2 secondes, sauf en cas de perte d'un message où ce temps dépasse alors les 60 secondes, qui correspond au délai d'envoi périodique de PIM join. Ces 2.2 secondes représentent une éternité comparé au 50 ms du temps de propagation, soit 22 fois l'aller-retour Strasbourg-Trondheim. La courbe de répartitions des temps d'adhésion (figure 1(c)) montre qu'un routeur met en moyenne 150 ms pour traiter la demande d'adhésion et la propager au routeur suivant.

Ce problème a été discuté pour IPv6 dans la liste de diffusion de m6bone [HOE 06] et aussi pour IPv4 [NAM 04]. De plus, ce délai n'apparaît que sur certains types de routeurs (par exemple les plus gros routeurs chez Cisco ou Juniper). Une analyse plus fine en utilisant des traces sur Cisco montre qu'un PIM join reçu n'est pas envoyé immédiatement mais est mis en attente pendant un certain temps ce qui permet d'agréger plusieurs messages et de limiter l'utilisation des ressources. Par conséquent, nous de-

2. « MRD6 » est un processus de routage multicast IPv6 pour Linux.
3. Les outils « multicat » (dont l'évolution se nomme maintenant « mcfirst ») et « ssmping » sont hébergés à l'adresse : « http ://www.venaas.no/multicast/ssmping/ ».



(a) FLID-SL local (b) FLID-DL local (c) WEBRC local

(d) FLID-SL m6bone (e) FLID-DL m6bone (f) WEBRC m6bone

Figure 2. *Résultats des tests pour FLID-SL, FLID-DL et WEBRC*

vons prendre en considération que le temps d'adhésion n'est pas négligeable et qu'il peut ainsi influer sur le comportement des mécanismes de contrôle de congestion.

5. Plateforme de tests des mécanismes de contrôle de congestion

Suite au constat de la section précédente et afin de vérifier le comportement des mécanismes de contrôle de congestion face à un temps d'adhésion variable, nous avons procédé à une série de tests en utilisant une application créée par nos soins qui implémente FLID-SL, FLID-DL et WEBRC. Chaque test se déroule en SSM IPv6 avec une source émettant à un débit maximal de 2 Mb/s. De plus, un lien d'une capacité d'1Mb/s est mis en place par traffic-shaping. Pour chaque type de contrôle de congestion, le test se résume à démarrer une session pendant 10 minutes et à observer l'évolution de la bande passante et du temps d'adhésion. Pour observer l'influence de ce dernier, les tests se sont déroulés dans deux environnements :

– Sur une maquette locale avec 2 routeurs et un temps d'adhésion court (< 50 ms).

– Sur le m6bone avec 17 routeurs et un temps d'adhésion long (≈ 2200 ms).

La figure 2 présente les résultats typiques du débit pour chaque type de contrôle de congestion. Ces résultats montrent des anomalies ainsi que les faiblesses de FLID-DL et WEBRC face à un temps d'adhésion conséquent. Nous allons donc analyser les modifications du comportement de FLID-SL, FLID-DL et WEBRC :

– Nous avons vu que FLID-SL utilise des canaux statiques. Les différents canaux

diffusent donc de manière constante et un récepteur doit s'abonner au prochain canal pour augmenter son débit. De ce fait un long temps d'adhésion ne provoque pas l'échec de l'abonnement au prochain canal, mais simplement une arrivée plus tardive des données. C'est pourquoi FLID-SL présente un comportement similaire en local (figure 2(a)) et sur le m6bone (figure 2(d)). Par contre cette lenteur du temps d'adhésion couplée à quelques pertes isolées nous font obtenir un débit inférieur sur le m6bone.

– FLID-DL (figure 2(b)) a un fonctionnement comparable à celui de FLID-SL en local (figure 2(a)), mais totalement différent sur le m6bone (figure 2(e)). Cette différence a pour origine l'utilisation de canaux dynamiques. En effet, un récepteur utilisant FLID-DL s'abonne au canal de base pour initier la session, avant de tenter de s'abonner au canal suivant pour augmenter son débit. Or ce canal est dynamique et devient muet au bout de TSD secondes, avec une valeur de 0.5 secondes par défaut pour TSD. De plus, le temps d'adhésion sur le m6bone est de 2.2 secondes et inclut un temps de branchement qui dépasse les TSD secondes. Ce temps de branchement supérieur à TSD fait qu'aucun paquet du canal demandé ne sera diffusé jusqu'au récepteur. Au changement de TSI, le récepteur se désabonne du canal et s'abonne au canal suivant. On se retrouve alors dans la même situation qu'au TSI précédent et avec un phénomène qui se répète indéfiniment et donne les résultats de la figure 2(e). Par conséquent un temps d'adhésion trop important rend impossible l'utilisation de FLID-DL, par un simple problème de démarrage.

– WEBRC présente également une meilleure utilisation de la bande passante en local (figure 2(c)) que sur la maquette du m6bone (figure 2(f)). Or, WEBRC utilise aussi des canaux dynamiques. Comme FLID-DL, un canal devient muet toutes les TSD secondes, mais par défaut TSD vaut 10 secondes pour WEBRC. Par conséquent, le problème de démarrage ne serait sensible que pour un très grand nombre de sauts (de l'ordre de 60 par exemple) et n'est pas visible sur nos tests. En outre, WEBRC présente une autre faiblesse relative au temps d'adhésion. En effet, nous avons vu que WEBRC utilise le temps d'adhésion pour le calcul de la bande passante disponible (voir équation 1). Par conséquent le phénomène constaté figure 2(f) provient du fait que les temps d'adhésion relevés sont beaucoup plus longs que le temps de propagation. Donc WEBRC calcule une bande passante disponible beaucoup plus faible que celle réellement disponible. De plus, ce calcul est réalisé pour prévenir des pertes, mais quand l'estimation est trop faible WEBRC ne reçoit plus d'informations sur l'état du réseau, car il ne provoque alors ni pertes, ni variations de propagation des paquets. En effet, le calcul de la bande passante disponible fait que WEBRC est plutôt préventif, alors que FLID-DL est réactif aux pertes.

Ainsi, nous avons mis en exergue deux nouveaux problèmes pour les mécanismes de contrôle de congestion qui sont le problème du démarrage (voir aussi [NEU 05]) et le problème du calcul de la bande passante disponible.

6. Solution proposée face aux problèmes détectés.

Nous avons vu que le temps d'adhésion est rarement représentatif du temps de

propagation. Pour les méthodes de contrôle de congestion, ce constat crée deux problèmes : le problème du démarrage et le problème d'estimation du débit disponible.
Le problème du démarrage est lié à l'utilisation de canaux dynamiques et apparaît quand le temps d'adhésion est supérieur au TSD du protocole utilisé. Pour déplacer ou supprimer cette limite, nous pouvons recourir à plusieurs solutions :

– Augmenter la valeur de TSD : Cette solution permet simplement de repousser la limite de notre problème. Cependant, il ne faut pas oublier que la valeur de TSD représente aussi le temps de réaction du protocole à une congestion. De ce fait augmenter la valeur de TSD signifie ralentir et dénaturer le mécanisme de contrôle de congestion.

– Effectuer des abonnements simultanés à plusieurs canaux : En effet, le premier canal dynamique se termine au bout de $1 * TSD$ secondes, le deuxième après $2 * TSD$ secondes, etc. S'abonner à plusieurs canaux simultanément permet donc d'augmenter la durée de diffusion des canaux. Mais cette méthode peut provoquer une congestion si par exemple un récepteur apparaît, réduisant ainsi la distance du routeur de branchement ce qui entraîne une brusque diminution du temps d'adhésion. La généralisation de ce cas consiste à envoyer les demandes d'adhésion en avance.

Au vu des problèmes engendrés, l'utilisation de ces solutions n'est pas recommandée. Le problème de l'évaluation du débit disponible provient de l'utilisation du temps d'adhésion par WEBRC. Cette variable n'est pas très représentative du temps de propagation alors qu'elle est sensée l'approximer. Une solution est donc de remplacer le temps d'adhésion dans l'équation 1 par une variable plus proche du temps de propagation. Malheureusement, obtenir un temps de propagation dans une communication unidirectionnelle est difficile. La méthode préventive de calcul du débit par WEBRC présente donc le risque de sous-utiliser la bande passante disponible.
Ces différentes difficultés nous mènent à proposer une solution couplant une source de type WEBRC, qui ne présente pas de problème de démarrage, avec un récepteur évaluant sa bande passante en réaction aux changements d'état du réseau.

7. Réactivité du récepteur

Nous voulons créer un « Récepteur Réactif (RR) » aux changements d'état du réseau qui sont observables par la détection de pertes et la variation du temps de propagation. Ainsi RR doit pouvoir atteindre un débit optimal sans utiliser de prédiction du débit disponible et en limitant les pertes. Le calcul de la variation du temps de propagation notée Δ nécessite d'ajouter une date d'estampillage notée $estampille$ dans les paquets émis par la source. Ensuite, le récepteur doit relever, pour chaque paquet reçu, la date d'arrivée du paquet noté $reception$. Ainsi entre deux paquets successifs, l'équation 2 permet de calculer Δ.

$$\begin{aligned} \delta_{source} &= estampille_{deuxieme\_paquet} - estampille_{premier\_paquet} \\ \delta_{recepteur} &= reception_{deuxieme\_paquet} - reception_{premier\_paquet} \\ \Delta &= \delta_{recepteur} - \delta_{source} \end{aligned} \quad [2]$$

Le calcul de Δ ne nécessite donc pas de synchronisation entre les horloges de la source et du récepteur. Cependant, dans le cas où $estampille_{deuxieme\_paquet} < estampille_{premier\_paquet}$, on observe les réactions suivantes :

– S'il y a un déséquencement entre deux paquets de deux canaux différents, alors il faut considérer cet événement comme une perte.

– Sinon ce déséquencement a déjà été interprété comme une perte pour le paquet précédent et ne sera pas à nouveau comptabilisé comme tel.

Maintenant que nous savons calculer Δ, l'algorithme de RR est relativement simple :

– Si le récepteur n'est pas déjà en train d'adhérer à un nouveau canal et que pendant une certaine durée notée $attente$ il ne connaît pas de pertes et que Δ est resté stable, alors il augmente son débit en s'abonnant à un nouveau canal.

– Sinon le récepteur ne fait rien et laisse son débit diminuer tout seul.

Nous définissons que Δ est stable si pour un $seuil$ égal au temps d'émission d'un paquet au débit actuellement reçu, nous vérifions que $\Delta \leq seuil$. De plus, comme la source diminue lentement son débit, il faut éviter les adhésions successives trop rapides. Ainsi nous empêchons le récepteur de s'abonner à un nouveau canal pendant 500 ms après chaque adhésion effective à un canal. Bien entendu, si le TSI change, alors le récepteur se désabonne du canal qui vient de devenir muet.

7.1. *Avantages de l'utilisation de la variation du temps de propagation*

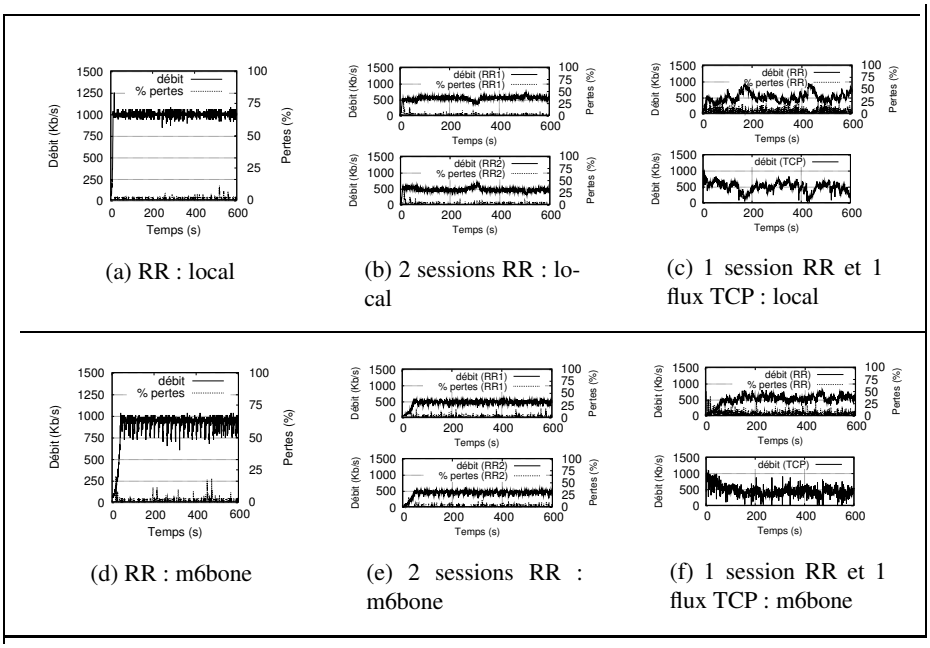
L'utilisation de la variation du temps de propagation supprime tous les paramètres néfastes que nous avons constatés pour le temps d'adhésion :

– Les paquets sont estampillés par la source. De ce fait, la variation du temps de propagation est indépendante de la construction de l'arbre multicast ainsi que de la position du noeud de branchement. Il n'y a pas d'influence du nombre de récepteurs sur la variation, contrairement au temps d'adhésion.

– Le deuxième point est que la variation du temps de propagation est calculée à partir de paquets estampillés par des dates. Ces paquets étant des paquets de données et non pas des paquets de signalisation, les routeurs ne les traiteront pas de manière singulière. Par conséquent, la variation du temps de propagation peut être mise en relation avec les variations du taux de remplissage des files d'attente de chaque routeur.

– En troisième point, la fréquence des paquets n'influe plus sur la justesse du résultat.

– En dernier point, la variation du temps de propagation est calculée à la réception de chaque paquet, contrairement au temps d'adhésion qui n'est relevé qu'à chaque nouvel abonnement à un canal.



(a) RR : local

(b) 2 sessions RR : local

(c) 1 session RR et 1 flux TCP : local

(d) RR : m6bone

(e) 2 sessions RR : m6bone

(f) 1 session RR et 1 flux TCP : m6bone

Figure 3. *Résultats des tests pour RR*

7.2. *Test du « Récepteur Réactif »*

Pour valider notre proposition, nous avons implémenté le mécanisme RR dans notre application et testé son efficacité avec les maquettes des tests précédents. Pour la source nous utilisons celle de WEBRC avec l'ajout des estampilles nécessaire au calcul de Δ. Le $seuil$ correspond au temps d'émission d'un paquet au débit reçu pendant les 500 dernières ms. La valeur du paramètre $attente$ a été évaluée empiriquement à 148 ms. Les résultats de la figure 3 mettent en évidence plusieurs points :

– Pour chacun des tests, RR utilise toute la bande passante disponible.

– La deuxième remarque est que RR converge très rapidement vers le débit optimal. En effet, alors qu'en local WEBRC converge en près de 100 secondes (figure 2(c)), il n'en faut que 10 pour RR (figure 3(a)). Sur le m6bone RR converge en 40 secondes (figure 3(d)), car il est ralenti par le temps d'adhésion.

– Un point négatif de RR est qu'il provoque des pertes plus importantes que WEBRC, car RR est un protocole réactif aux pertes contrairement à WEBRC qui essaye de les prédire. Mais ce taux de pertes reste tout de même inférieur à 1% pour tous les tests n'utilisant pas TCP et est inférieur à 5% pour les tests mêlant RR et TCP. Quand RR détecte une importante variation positive du temps de propagation, il n'augmente pas son débit car il sait qu'un abonnement provoquerait de nombreuses pertes. Malheureusement, la source de WEBRC réduit son débit lentement et face à cette réduction notre prédiction est trop tardive pour éviter les pertes.

– Les figures 3(b) et 3(e) montrent deux instances concurrentes de RR, alors que les figures 3(c) et 3(f) montrent une instance de RR concurrente à une instance de TCP. Ces résultats montrent que RR est équitable envers lui-même et envers TCP dans le sens où aucune instance n'usurpe continuellement la bande passante de l'autre. Cependant la rapidité de convergence vers le débit le plus équitable pourrait être améliorée en utilisant une source dont le débit diminue plus rapidement.

Pour réduire le taux de pertes et améliorer l'équité, nous envisageons de créer une source utilisant des canaux à dynamicité rapide et ne provoquant pas de problème au démarrage. RR est donc un mécanisme de contrôle de congestion prometteur, qui demande une étude plus approfondie et la création d'une régulation efficace du débit de la source.

8. Conclusion

Cet article a permis de montrer que le temps d'adhésion est une variable complexe dont les nombreuses composantes sont rarement représentatives du temps de propagation. Nous avons testé et relevé que ce temps pouvait être particulièrement long. Après analyse et recoupement des informations recueillies, nous avons conclu que certains routeurs des grands équipementiers présentent une latence de 150 ms en moyenne lors d'une demande d'adhésion. Ce n'est pas la mise à jour de la table de routage multicast qui est concernée par cette latence, car il s'agit uniquement d'une attente avant propagation du message d'adhésion.
Suite à ces observations, nous avons mis à l'essai trois méthodes de contrôle de congestion multicast en utilisant des maquettes présentant de grandes différences de temps d'adhésion. Les résultats de ces tests ont mis à jour deux problèmes dus aux lenteurs du temps d'adhésion. Le premier problème est lié au démarrage des protocoles à canaux dynamiques, alors que le second est un problème d'estimation du débit disponible pour WEBRC. Face à ces problèmes, nous avons proposé une première solution utilisant une source WEBRC avec un récepteur réactif aux pertes et à la variation du temps de propagation du réseau. Finalement, nous avons testé ce mécanisme qui permet une utilisation optimale de la bande passante. En perspective, nous envisageons de créer une source disposant d'une meilleure courbe de réduction du débit, ainsi que l'utilisation d'une fenêtre d'émission virtuelle avec un comportement comparable à celui de la fenêtre d'émission de TCP.

9. Bibliographie

[BYE 02] BYERS J. W., HORN G., LUBY M., MITZENMACHER M., SHAVER W., « FLID-DL : Congestion Control for Layered Multicast », *IEEE journal on selected areas in communications*, vol. 20, NO.8, 2002, p. 1558–1570.

[BYE 06] BYERS J. W., KWON G.-I., LUBY M., MITZENMACHER M., « Fine-Grained Layered Multicast with STAIR », *IEEE/ACM Transactions on Networking*, vol. 14, NO.1, 2006, p. 81–93.

[HOE 06] HOERDT M., OWENS B., VENAAS S., ASAEDA H., GALL A., BHASKAR N., SAVOLA P., « [m6bone] Join delay tree setup time, Why is it so long on Cisco/Juniper ? », IPv6 Multicast mailing-list – m6bone@ml.renater.fr, 2006, https ://listes.renater.fr/sympa/arc/m6bone/2006-01/msg00041.html.

[LEG 00] LEGOUT A., BIERSACK E. W., « PLM : Fast Convergence for Cumulative Layered Multicast Transmission Schemes », *Proc. of ACM SIGMETRICS'2000*, Santa Clara, California, USA, 2000, p. 13–22.

[LUB 00] LUBY M., VICISANO L., HAKEN A., « Reliable Multicast Transport Building Block : Layered Congestion Control », Internet draft : expired, November 2000, Remote Working Group - IETF.

[LUB 02a] LUBY M., GOYAL V., SKARIA S., HORN G., « Wave and Equation Based Rate Control Using Multicast Round Trip Time », *ACM SIGCOMM'02*, 2002.

[LUB 02b] LUBY M., VICISANO L., GEMMELL J., RIZZO L., HANDLEY M., CROWCROFT J., « The use of Forward Error Correction in Reliable Multicast », RFC : Informational n° 3453, December 2002, Network Working Group - IETF.

[LUB 04] LUBY M., GOYAL V., « Wave and Equation Based Rate Control building block », RFC : Experimental n° 3738, April 2004, Network Working Group - IETF.

[LUB 05] LUBY M., « Basic Forward Error Correction (FEC) Schemes - draft-ietf-rmt-bb-fec-basic-schemes-revised-00 », Internet draft : expires : January 14, 2006, July 2005, Remote Working Group - IETF.

[NAM 04] NAMBURI P., SARAC K., ALMEROTH K. C., « SSM-Ping : A Ping Utility for Source Specific Multicast », *3rd IASTED International Conference on Communications, Internet, and Information Technology (CIIT)*, St. Thomas, US Virgin Islands, USA, 11 2004, p. 63–68.

[NEU 05] NEUMANN C., ROCA V., « Impacts of the Startup Behavior of Multilayered Multicast Congestion Control Protocols on the Performance of Content Delivery Protocols », *IEEE 10th International Workshop on Web Content Caching and Distribution*, 9 2005.

[PAD 98] PADHYE J., FIROIU V., TOWSLEY D., KUROSE J., « Modeling TCP Throughput : A Simple Model and its Empirical Validation », *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, 1998, p. 303–314.

[PAI 04] PAILA T., LUBY M., LEHTONEN R., ROCA V., WALSH R., « FLUTE - File Delivery over Unidirectional Transport », RFC : Experimental n° 3926, October 2004, Network Working Group - IETF.

[PUA 03] PUANGPRONPITAG S., « Design and Performance Evaluation of Multicast Congestion Control for the Internet », PhD thesis, School of Computing, University of Leeds, 11 2003.

[QUI 01] QUINN B., ALMEROTH K., « IP Multicast Applications : Challenges and Solutions », RFC : Informational n° 3170, 9 2001, Network Working Group - IETF.