LCC : Un réseau de recouvrement multipoint passant à l'échelle

Mohamed Ali Kaafar, Thierry Turletti, Walid Dabbous

LCC : Un réseau de recouvrement multipoint passant à l'échelle

Mohamed Ali Kaafar — Thierry Turletti — Walid Dabbous

INRIA Sophia Antipolis
2004 route des Lucioles – B.P. 93
06902 Sophia Antipolis Cedex
{mkaafar, dabbous, turletti}@sophia.inria.fr

RÉSUMÉ. *Les propositions récentes de construction de réseaux de recouvrement (overlay) pour la transmission multipoint ont démontré l'importance d'exploiter les informations de la topologie du réseau sous-jacent. Toutefois, ces propositions reposent souvent sur des processus de raffinement incrémentaux pour améliorer les performances du système. Ces approches ne passent pas de ce fait à l'échelle, induisant un surcoût de communication élevé, et nécessitent un temps de convergence important avant d'atteindre une structure stable. Dans cet article, nous proposons un algorithme de localisation qui dirige graduellement les nouveaux venus vers les nœuds les plus proches sans pour cela induire un surcoût élevé. En nous basant sur cette localisation, nous proposons un réseau overlay, nommé "LCC", à la fois robuste, passant à l'échelle et prenant en compte la topologie physique. Nous avons mené des expérimentations réelles sur PlanetLab ainsi que des simulations afin d'évaluer les performances de LCC. Les résultats prouvent d'une part que le processus de localisation nécessite des ressources modestes en termes de délais et de bande passante, et d'autre part que LCC est un overlay efficace pour supporter des applications multipoints à très grande échelle.*

ABSTRACT. *Recent topology-aware proposals in multicast overlay construction often rely on incremental and periodic refinements to improve the system performance. These approaches are therefore neither scalable, as they induce high communication cost due to refinement overhead, nor efficient because long convergence time is needed to stabilize the structure. In this paper, we propose a highly scalable locating algorithm that gradually directs newcomers to a set of their closest nodes without inducing high overhead. On the basis of this locating process, we build a robust and scalable topology-aware overlay scheme, called LCC. We conducted both simulations and PlanetLab experiments to evaluate the performance of LCC. LCC demonstrates promising performance to support large scale multicast applications.*

MOTS-CLÉS : *Réseaux de recouvrement, transmission multipoint, réseaux pair-à-pair, topologie réseau, performance.*

KEYWORDS: *Overlay networks, multicast, peer-to-peer networks, network topology, performance.*

1. Introduction

Un des facteurs majeurs au succès des réseaux de recouvrement (*overlay*) est la facilité de déploiement et de mise à jour des services overlays. En particulier, plusieurs overlays permettent de supporter un service de dissémination des données sous forme de transmission multipoint, sans pour autant nécessiter le déploiement du service multipoint IP natif. Cependant, ce type de service multipoint applicatif souffre souvent de faibles performances, de problèmes de passage à l'échelle et de coût. Ces problèmes s'aggravent lorsque la construction de l'arbre de transmission des données ignore la topologie et les caractéristiques des liens du réseau sous-jacent. Les propositions récentes pour la construction d'overlays multipoints ont démontré l'importance primordiale d'exploiter des informations provenant du réseau physique sous-jacent. Cependant, nous démontrerons dans cet article que des barrières persistent quant à la qualité de service offerte par les overlays multipoints actuels, en l'occurrence des problèmes de passage à l'échelle et d'efficacité :

1) Bien que les protocoles décentralisés aient été conçus pour passer à l'échelle, en ne reposant pas sur une connaissance globale de la topologie, ils sont souvent confrontés au problème de surcoût de communication élevé, induit par leur processus de raffinement. En effet, dans de tels protocoles, [CH00] [HE02] [LI03] [TA05], les nœuds maintiennent des positions relatives par rapport à la source de l'arbre de transmission. Périodiquement, chaque nœud essaye d'améliorer sa position en recherchant un meilleur parent, i.e. un nœud non descendant qui lui fournirait un meilleur délai à la source. Ce type de processus de raffinement, ne permet pas à la structure overlay de passer à l'échelle, d'autant plus si l'on considère une certaine dynamique au niveau des adhésions des nœuds à l'overlay, ou au niveau des conditions du réseau sous-jacent. Un coût supplémentaire de message de contrôle sera induit lors des opérations périodiques de maintenance et de réparation des structures. D'un autre côté, une fréquence plus élevée d'émission des messages de contrôle sera nécessaire pour transposer des variations des caractéristiques de la topologie physique vers la topologie virtuelle.

2) Les utilisateurs assistant à une vidéo conférence ou à une diffusion d'événement s'attendent à une qualité acceptable dès lors qu'ils rejoignent la session multipoint. Puisqu'un arbre de transmission overlay multipoint est typiquement étudié dans le but de minimiser le délai moyen induit observé par les récepteurs, nous considérerons que cet arbre est efficace si son délai moyen induit est inférieur à une valeur seuil. Nous nous attendons de ce fait à ce que les approches qui se basent sur des raffinements incrémentaux nécessitent un long délai avant que l'arbre de transmission ne converge vers une structure efficace.

Dans cet article, nous nous proposons de fournir une solution pratique pour le support d'un service multipoint efficace et à grande échelle. Dans un premier temps, nous proposons un processus simple et précis pour la détection des voisins et la localisation. L'idée principale est d'exploiter les nœuds déjà existants dans l'overlay afin de suggérer d'éventuels voisins au nouveau venu. Ce dernier envoie des requêtes de localisation de proche en proche vers les candidats suggérés, en affinant à chaque fois son positionnement dans le réseau sous-jacent. Un des atouts de ce processus de localisation est qu'il n'utilise ni les systèmes de coordonnées virtuelles, ni de mesures vers des bornes balises fixes. Il vise ainsi à la fois à passer à l'échelle et à être précis.

Reposant sur les résultats du processus de localisation, nous construisons dans une seconde étape un overlay hiérarchique, à base de groupes, à la fois robuste aux différents changements, prenant en compte la topologie sous-jacente et passant à l'échelle. Nous proposons des mécanismes pro-actifs pour prévenir les pannes des dirigeants de groupes, et pour s'adapter aux changements de topologies. Le passage à l'échelle se fait en réduisant les flux de contrôle et notamment la bande passante nécessaire pour la maintenance de l'overlay. Le schéma de construction de l'overlay multipoint ainsi proposé, se nomme LCC ("Locate, Cluster and Conquer") pour Localiser, Regrou-

per et Conquérir. Intuitivement, la localisation effectuée avant de joindre l'overlay, puis le regroupement de nœuds proches, devrait permettre à l'arbre de transmission multipoint de converger rapidement et de réduire le surcoût dû à la maintenance incrémentale. Cependant, ces perfectionnements pourraient être mitigés par un surcoût éventuel du processus de localisation.

En tenant compte de ces considérations, nous avons évalué le schéma LCC en utilisant deux méthodes complémentaires d'évaluation à savoir, des simulations et des expérimentations sur la plateforme PlanetLab. Les résultats prouvent que LCC engendre un faible surcoût de communication lors de la phase de localisation, ainsi que durant la session multipoint. Comparé à d'autres protocoles, se connectant initialement d'une manière aléatoire ou tenant en compte de la topologie, LCC permet un temps de convergence plus court et une faible fréquence d'ajustement des liens overlay. Enfin, LCC est capable de construire un arbre multipoint efficace, en particulier pour des overlays de grandes tailles.

Cet article est structuré comme suit : La section 2 présente les travaux concernant la construction d'overlays multipoints. La section 3, introduit le mécanisme LCC, et détaille les deux processus de localisation et regroupement. Les expérimentations et simulations sont discutées dans la section 4. LCC est comparé à diverses approches précédemment proposées. La section 5 conclut cet article.

2. Etat de l'art

Un trés grand d'intérêt a été porté à la construction d'overlays afin de fournir le service multipoint au niveau applicatif. Les différentes contributions peuvent être classées en approche à base de routeur overlay et approche pair-à-pair (P2P).

Dans les approches "routeurs overlay", comme OMNI [BA03] et TOMA [LA05], des serveurs sont installés à travers le réseau en agissant comme des routeurs multipoints de niveau applicatif. Les données sont transmises de la source vers un ensemble de récepteurs sur l'arbre multipoint composé des serveurs overlay. Cette approche a été conçue pour passer à l'échelle, puisque les récepteurs reçoivent les données à partir des serveurs, diminuant ainsi la demande en bande passante au niveau de la source. Cependant, elle nécessite le déploiement de serveurs coûteux dédiés à cette tâche.

L'approche P2P ne requiert aucune ressource supplémentaire. Plusieurs propositions permettent de gérer des sessions multipoints avec des groupes réduits. Narada [CH00], MeshTree [TA05], et Hostcast [LI03] sont des exemples d'algorithmes "Mesh-first" distribués, où les nœuds s'arrangent en maille au dessus de laquelle un algorithme de routage permet de dériver un arbre de transmission. Ces protocoles se basent sur des raffinements incrémentaux qui rajoutent et suppriment tour à tour des liens de la maille afin d'optimiser une fonction d'utilité. Bien que ces protocoles offrent des propriétés de robustesse (grâce à la structure maillée), ils ne passent pas à l'échelle, à cause du surcoût excessif généré lors des processus de raffinement. L'objectif de LCC est de localiser un nouveau venu avant qu'il ne se joigne à l'overlay, pour lui permettre de réduire le nombre de processus de raffinements par la suite.

D'autres protocoles "Tree-First", comme ZigZag [TR03] et NICE [BA02], sont des protocoles prenant en compte la topologie et se basant sur le regroupement. Ils ont été conçus pour supporter de grandes sessions multipoints, avec des applications à faible débit. Ces protocoles ne considèrent pas en revanche les capacités individuelles de transmission (fan-out) de chaque nœud. Ils délimitent par contre la capacité totale de chaque groupe de l'overlay par un paramètre "taille des groupes". En particulier, puisque ces protocoles ne considèrent que les délais entre nœuds comme critère de sélection des dirigeants de groupes, ils peuvent rencontrer des problèmes si ces dirigeants ont des capacités insuffisantes. D'autres protocoles exploitent des informations de topologie au niveau des AS (Systèmes Autonomes) [SO04] ou au niveau des rou-

teurs [KW02], pour construire des réseaux overlay efficaces. Ces approches supposent une certaine assistance de la part de la couche IP (des routeurs émettant des messages ICMP, ou un accès aux informations BGP) qui pourrait être problématique. LCC ne requiert aucune assistance de la part d'entités n'appartenant pas à l'overlay.

L'approche de regroupement par balises fixes (landmarks) est un concept général pour construire des overlays qui tiennent compte de la topologie. Ratnasamy et al. [RA02] utilise cette approche pour construire un overlay CAN-multipoint. Avant de se joindre à la session, un nouveau venu doit mesurer sa distance vers chaque balise, puis ordonner les balises selon ses mesures de distance. L'idée principale est que les nœuds avec des ordonnancements de balises voisins sont probablement proches l'un de l'autre dans la topologie physique. Un des inconvénients immédiats d'une telle approche est que sa précision dépend du nombre de balises déployées, ainsi que de leur distribution à travers le réseau. De plus, le fait de nécessiter des balises fixes connues par tous les participants, rend cette approche inadaptée pour des réseaux dynamiques.

3. L'algorithme LCC

LCC est composée de deux processus : la localisation et le regroupement.

Le **processus de localisation** vise à diriger les nouveaux venus vers le groupe le plus "proche" avant qu'il ne reçoive des données sur l'arbre de transmission. Un nouveau venu initie le processus en émettant une requête de localisation à un dirigeant de groupe choisi aléatoirement. Selon la précision de sa localisation, ce dernier choisit des dirigeants de groupes (que nous dénoterons *nœuds invités*) qu'il considère comme étant proches du nouveau venu. Il invite alors ces nœuds à sonder le nouveau venu, et mémorise chaque réponse. Il suggère ensuite au nouveau venu les nœuds les plus proches possibles. En émettant itérativement des messages "Requête de localisation" au nœud suggéré le plus proche (dénommé *nœud relais*), le nouveau nœud est capable de localiser de proche en proche ses nœuds voisins dans la topologie physique. Le processus se termine en proposant un ou plusieurs dirigeants de groupes proches.

En regroupant les nœuds proches d'un unique dirigeant au sein d'un même groupe (cluster), nous nous attendons à ce que les membres soient proches les uns des autres. Cela permet ainsi des messages intra-groupe à faible coût. Le **processus de regroupement** est initié par chaque nœud, une fois le processus de localisation terminé. Une distance maximale, R_{max}, caractérise l'intervalle dans lequel les nœuds sont considérés proches. Cet intervalle, appelé *portée* du dirigeant de groupe, définit le critère de regroupement. Durant ce processus, un nœud décide à quel niveau de l'overlay se joindre. S'il crée son propre groupe, il se joint au niveau supérieur (haut niveau) de la topologie et commence la construction d'une maille inter-groupes. Sinon, il devient membre d'un groupe et se joint à la maille intra-groupe. La figure 1 présente une vue

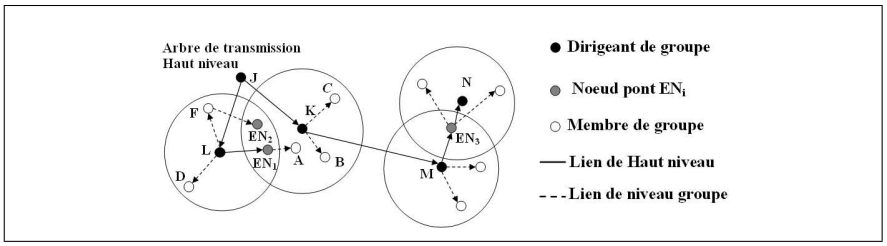

Figure 1. *La hiérarchie à deux niveaux de LCC*

générale du mécanisme LCC.

Un nœud pouvant être à la portée de plusieurs dirigeants à la fois, il peut être membre de plusieurs groupes. Ce type de nœud est appelé *nœud pont*. Nous exploiterons les nœuds ponts afin d'améliorer l'efficacité de l'overlay. En fait, les nœuds ponts sont, comme les dirigeants de groupes, autorisés à joindre la maille inter-groupe au haut niveau. Le rôle principal d'un nœud pont est de permettre (si les contraintes de capacité individuelle ne sont pas violées) aux membres des groupes de dériver leur arbre de transmission en considérant les nœuds ponts comme sources alternatives (en plus du dirigeant) connectées à la topologie supérieure. De plus, ces nœuds contribuent à la robustesse de l'overlay dans le cas d'échec des dirigeants. Bien que les nœuds ponts soient attachés à plusieurs mailles de plus d'un groupe, ils ne reçoivent pas les données plus d'une fois. En effet, puisque chaque nœud pont dérive un unique arbre de transmission à partir de l'une de ses mailles intra-groupe existantes, il est ainsi fils dans cet arbre particulier. D'un autre côté, il pourrait être parent dans plusieurs autres arbres de transmission dérivés dans d'autres groupes.

Notons que LCC ne spécifie pas un nouveau protocole pour la construction d'arbre de transmission : n'importe quel protocole existant peut être greffé au dessus de LCC. LCC est construit en utilisant MeshTree [TA05] sur chacun des niveaux intra et inter-groupe. MeshTree implante l'arbre de transmission dans une maille contenant plusieurs liens de faible coûts. La structure "plate" de MeshTree est au départ aléatoirement construite, et se base sur des opérations périodiques d'ajout/suppression de liens. LCC, en revanche, construit initialement un overlay tenant en compte de la topologie sur la base des processus de localisation et de regroupement. Les nœuds de haut niveau agissent alors comme des nœuds MeshTree particuliers, où les autres groupes représentent des voisins dans l'arbre de transmission dérivé (voir figure 1).

3.1. *Le processus de localisation*

Nous adoptons une stratégie de positionnement similaire à celle de meridian [WO05] afin d'organiser les nœuds dans des niveaux en fonction d'une métrique de distance. Typiquement, la distance entre deux nœuds est le délai réseau d'aller retour (RTT). Chaque nœud maintient une liste de nœuds dans son système de localisation. Ce système est composé d'un ensemble de niveaux, ne se recouvrant pas, et exponentiellement croissants. Il est représenté par des intervalles $[r_i, r_{i+1}[$, où $r_i = \alpha\ e^{i-1}$ pour $i >= 1$ et $r_0 = 0$ (figure 2). Chaque nœud mesure sa distance vers un ensemble de

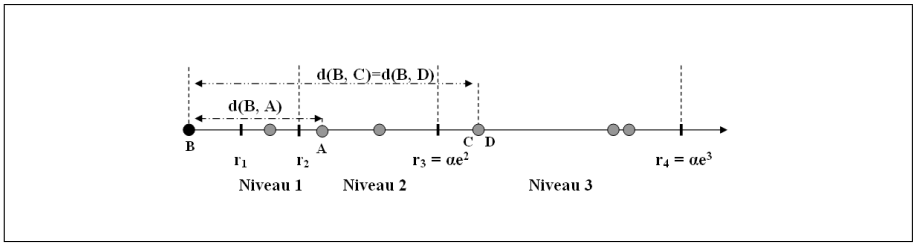

Figure 2. *Le sytéme de localisation du nœud B*

nœuds dont il a connaissance et leur affecte à chacun une position dans le niveau correspondant de son système de localisation. A titre d'exemple, si la distance mesurée, d satisfait $r_i \leq d < r_{i+1}$, le nœud est positionné dans le $i^{ème}$ niveau. Tous les nœuds considérés dans le système de localisation sont des dirigeants de groupes. Dans ce qui suit, nous allons décrire les opérations de localisation.

3.1.1. *Initialisation et requête de localisation*

Initialement, un nœud A, doit contacter une entité, appelée point de Rendez-Vous, définissant la session multipoint afin d'obtenir l'identité d'un nœud aléatoirement choisi, B, ayant déjà rejoint le système. A mesure sa distance vers B, $d(A,B)$ et affecte à ce dernier un niveau, i, dans le système de localisation. Si A est à la portée de B ($d(A,B) <= R_{max}$), le critère de regroupement est vérifié et le processus de localisation se termine. A émet alors une requête pour rejoindre le groupe de B. Sinon, A émet une requête de localisation à B. En recevant cette requête, le nœud B demande simultanément aux nœuds invités de mesurer leur distance de A. Ces nœuds invités reportent leur résultat au nœud relais (en l'occurrence B actuellement). Si un nœud invité est plus proche du nouveau venu que le nœud relais, il est considéré comme candidat. Une liste identifiant les nœuds candidats est envoyée ainsi par le nœud relais au nouveau venu A. Parmi cette liste, A initie des processus visant à se joindre aux groupes dont les dirigeants satisfont le critère de regroupement. À défaut, A ré-initie son processus de localisation avec chaque nœud candidat par ordre croissant de distances. La liste est mise à jour à chaque réponse d'un nœud relais. Cette procédure se répète jusqu'à ce que le nouveau venu détecte un dirigeant dans sa portée. Finalement, il est nécessaire de fixer un critère d'arrêt pour que le processus se termine en un temps limité en répétant la procédure C fois. Si l'algorithme se termine sans satisfaire le critère de regroupement, A crée son propre groupe.

3.1.2. *Le processus de localisation sélectif*

Durant la requête de localisation, chaque nœud relais doit collecter des informations de distance des nœuds invités qu'il considère proche du nouveau venu. Il choisit alors parmi eux une liste de candidats qu'il émet au nœud en phase de localisation. Dans cette sous-section, nous répondons à la question suivante : comment les nœuds invités sont-ils choisis par le nœud relais ?

Une solution intuitive serait que le nœud relais invite tous les nœuds se trouvant dans le même niveau, ainsi que dans les niveaux adjacents, du nouveau venu. Pour établir une référence, nous considérons cette solution que nous appellerons localisation "*non sélective*". Malgré un avantage de simplicité, cette solution pourrait induire un surcoût de communication élevé. En effet, bien que dans le même niveau ou dans les niveaux adjacents que celui du nouveau venu, certains nœuds ne devraient pas être considérés pour mesurer leur distance vers le nouveau venu, puisqu'ils peuvent ne pas être plus proches du nouveau venu que le nœud relais.

Nous introduisons le critère de sélection afin de réduire le nombre de mesures inutiles durant le processus de localisation. La "*localisation sélective*" consiste à inviter uniquement des nœuds représentatifs à mesurer leur distance vers le nouveau venu. Les nœuds suffisamment proches d'un nœud représentatif, ne sont pas invités à effectuer des mesures. Moins de mesures de distances effectuées signifie moins de surcoût relatif à ces mesures et moins de messages de contrôle.

La notion de proximité à un nœud représentatif est définie par une valeur seuil γ_i, fonction de la distance entre le nouveau venu et le nœud relais, d. Si le nœud est proche d'une frontière de niveau ou du nœud relais, ce dernier devrait utiliser une sélection fine et une petite valeur de γ_i devrait être utilisée. Dans le cas contraire, le nœud relais devrait utiliser un plus grand γ_i. Dans notre algorithme, nous choisissons :

$$\gamma_i = \frac{|d - r_i|}{r_{i+1}} \times d$$

Les nœuds maintiennent pour chaque niveau i une matrice carrée M_i, représentant les distances connues jusque là entre les nœuds du niveau i, et des niveaux adjacents $i+1$

et $i-1$. Les valeurs de M_i sont affectées au fur et à mesure qu'elles sont découvertes à travers les requêtes de localisation des autres nœuds. Si une distance n'est pas connue, elle est fixée à une valeur assez large pour éliminer le nœud concerné par le processus de sélection. Chaque élément $M_i(j,k) = d(N_j^i, N_k^i)$ dans M_i correspond à la distance entre les nœuds N_j^i et N_k^i. La j^{eme} colonne dans M_i représente les distances connues entre le nœud N_j^i et les nœuds de niveau i ou ceux des niveaux adjacents. L'algorithme de sélection exécuté au niveau d'un nœud relais est présenté dans Algorithme 1, et

Algorithm 1 Selection

Require: Distance
Ensure: List of representative nodes to query

$Level \leftarrow Assign\_Level(Distance)$
$Candidates \leftarrow Search\_Nodes(Level)$
$S^{Level} \leftarrow Get\_Distance\_Matrix(Candidates)$
$Threshold \leftarrow \frac{|Distance - r_{Level}| \times Distance}{(r_{Level+1})}$
repeat
  $j \leftarrow Random(Dimension(S^{Level}))$
  $V \leftarrow Extract\_Row(S^{Level}, j)$
  for $i \in Dimension(S^{Level})$ **do**
    if $V(i) < Threshold$ **then**
      $Represented \leftarrow Represented \cup Index\_to\_Node(i)$
    end if
  end for
  $Representative \leftarrow Representative + Index\_to\_Node(j)$
  $S^{Level} \leftarrow S^{Level} \setminus Columns(Represented)$
until $Elements(S^{Level}) = Representative$
Return $Representative$

pourrait être décrit comme suit :

Chaque nœud relais choisit aléatoirement un nœud, N_j^i, du niveau i ou des niveaux adjacents. Si $M_i(j,k) = d(N_j^i, N_k^i)$ est inférieure à γ_i, le nœud N_k^i est représenté par N_j^i. Les nœuds sélectionnés sont représentés par une matrice, soit S_i, initialement égale à M_i. À chaque itération du processus de sélection, S_i est réduite des colonnes de M_i dont les nœuds peuvent être représentés par un nœud sélectionné N_j^i. L'algorithme de sélection se termine quand S_i ne contient plus que des distances de nœuds représentatifs.

3.2. *Le processus de regroupement*

Dans cette section, nous mettons l'accent sur les mécanismes utilisés afin d'améliorer le passage à l'échelle et la robustesse de l'overlay. En particulier, nous décrivons brièvement le processus d'élection des dirigeants, ainsi qu'un algorithme proactif prévenant les pannes des dirigeants et les changements de topologies. Les détails concernant la création de groupes, la manière de joindre un groupe et la maintenance de l'overlay sont détaillés dans [KA06].

3.2.1. *Élection des dirigeants*

L'élection de dirigeants est basée sur la valeur de vecteurs de priorités (PV) utilisés par les nœuds pour maintenir un rang entre eux mêmes. Un PV est défini comme :

$PV =< f^{max}, \frac{1}{DL}, T, \frac{1}{CD}, Migrated >$, où DL est le temps de latence perçu par le nœud dans l'arbre de transmission intra-groupe, CL dénote la distance minimale entre le nœud et le dirigeant d'un groupe différent du sien, T représente le temps passé par le nœud dans l'overlay et $Migrated$ est un booléen indiquant si le dirigeant est (encore) inclus dans la portée du nœud. La valeur de vecteurs de priorités est calculée par combinaison linéaire des 4 premiers composants de PV, avec des poids décroissants. Les priorités sont utilisées pour trier les nœuds éligibles appropriés. Les nœuds partagent leur PV par inclusion dans les messages d'entretien de l'overlay "Keep Alive". En mettant à jour les valeurs de PV, les nœuds s'affectent des priorités croissantes. Ils construisent ainsi un plan de secours pro-actif, où chaque nœud maintient dans un cache local une liste triée des nœuds éligibles à devenir des dirigeants. Dans un environnement dynamique, un dirigeant peut quitter l'overlay à tout moment. Dans ce cas, les nœuds LCC le détectent car ils ne reçoivent plus de messages "Keep Alive" de sa part, et un nouveau dirigeant est automatiquement élu.

3.2.2. *Topologie dynamique des groupes*

Dans ce paragraphe, nous discutons le comportement de LCC en cas de migration des membres en dehors de leur groupe, pour cause d'élection d'un nouveau dirigeant ou de changements dans la topologie physique. Nous distinguons trois états : (1) **Etat stabilisé** : où le dirigeant est inclus dans la portée de chaque membre. (2) **Etat temporaire** : où au moins un des nœuds a migré en dehors de son groupe. (3) **Etat de récupération** : où durant l'état temporaire, tous les nœuds ayant migré se sont mutuellement détectés et commencent à évoluer vers un état stabilisé.

Nous présentons un algorithme permettant aux nœuds ayant migré de collaborer afin de créer de nouveaux groupes appropriés après l'état temporaire. Plus précisément, les nœuds vérifient à chaque mise à jour de leur cache local, si le dirigeant actuel est dans leur portée ou pas. Si non, ils le marquent comme groupe voisin dans leur PV en affectant $Migrated$ = 1. Pour chaque PV reçu, un nœud ayant migré met à jour un ensemble des autres nœuds migrants. Le premier nœud (toujours selon les PV) de cet ensemble initie le processus en créant un nouveau groupe et en émettant un message "Recovering Request" aux prochains nœuds migrants de la liste. La requête contient les nœuds qui sont déjà dans la portée du nœud émetteur de la requête. Ainsi, chaque nœud est capable, d'après les précédentes requêtes, de déterminer les nœuds migrants qui peuvent encore être des dirigeants. Si un nœud est inclus dans la portée du nœud émetteur du "Recovering Request", il envoie un ACK positif pour rejoindre son groupe et retourne à un état stabilisé. Un nœud qui envoie un ACK négatif, vérifie s'il a été contacté par tous les nœuds éligibles migrants, le précédant dans le cache local. Si c'est le cas, il devient dirigeant de groupe et initie à son tour sa propre procédure de récupération. L'algorithme se termine en contactant le dernier nœud migrant. Le nœud (nouveau dirigeant) informe alors ses (nouveaux) groupes voisins, ainsi que son ancien dirigeant du partitionnement de groupe [KA06].

4. Evaluation des Performances

Pour évaluer le mécanisme LCC, nous avons effectué des simulations ainsi que des expérimentations PlanetLab [PL]. Alors que le but des simulations est d'évaluer l'efficacité des techniques proposées pour des overlays à grande échelle, les expériences sur la plateforme PlanetLab visent à illustrer les performances du système dans un environnement réel particulier.

4.1. *Mise en place des simulations et expérimentations*

En utilisant le générateur de topologie Internet BRITE [ME01], nous avons simulé des réseaux de 10^4 nœuds. La distribution de bande passante des pairs Gnutella observé par Saroiu et al. dans [SA02] a été utilisée comme modèle. Nous avons mo-

délisé : (1) les arrivées des nœuds dans l'overlay par une distribution de Poisson de taux moyen égal à 60 nœuds par pas de simulation (tick), et (2) la durée de vie dans l'overlay par une distribution exponentielle de paramètre égal à 0,01.

Nous avons testé LCC sur un ensemble de 212 nœuds PlanetLab (90 nœuds aux Etats Unis, 90 nœuds en Europe et 32 en Asie). La source "planetlab1.cs.cornell.edu" génère un flux de données à 70 KB/s. Les nœuds se joignent à l'overlay avec un taux moyen de 2 nœuds par seconde. Nous avons aussi implanté LCC dans une librairie et écrit des "Wrappers" pour les applications IP-multipoint natif (vic, rat et vlc)[1].

4.2. *Métriques et comparaison*

Nous évaluons le mécansime LCC en termes de (1) passage à l'échelle, en étudiant le coût des messages de contrôle et en observant la fréquence d'ajustements des liens ; (2) efficacité, en mesurant la moyenne de Pénalité Relative en Délai (Average Relative Delay Penalty : $ARDP$), qui est définie comme le ratio moyen entre le délai moyen overlay(d') et le délai du plus court chemin dans le réseau sous-jacent (d) de s jusqu'à tous les autres nœuds : $\frac{1}{N-1}\sum_{i=1}^{N-1}\frac{d'(s,i)}{d(s,i)}$, où N est le nombre de nœuds dans l'overlay. Ensuite, en considérant que l'arbre de transmission converge ou devienne "efficace" lorsque $ARDP$ est inférieur à une valeur seuil (égale à 2), nous étudions le temps de convergence. Nous quantifions aussi l'efficacité de LCC en observant le nombre de copies de paquets identiques sur un même lien physique, appelé stress sur le lien.

En utilisant ces métriques, nous comparons LCC à divers overlays. Pour des protocoles se basant sur des opérations de raffinements périodiques, nous expérimentons une variante de LCC, neutralisant le processus de localisation et fixant la valeur de la portée de chaque nœud à 0. Nous émulons de ce fait le comportement du protocole MeshTree. Nous appelons cette variante "Flat MeshTree". Dans les simulations, nous comparons aussi LCC à deux structures overlay multipoint précédemment proposées : OMNI [BA03] comme approche à base d'infrastructure (routeurs overlay), et ZigZag [TR03] comme une approche hiérarchique tenant en compte de la topologie.

4.3. *Coût en communications de contrôle*

Nous avons mené des simulations afin d'évaluer le surcoût du trafic de contrôle dans l'overlay et analysé le comportement du protocole dans des overlays de grandes tailles. Nous supposons une taille de 40 octets pour l'entête de chaque paquet IP, et nous mesurons le trafic total résultant des messages de contrôle. La figure 3 montre le surcoût moyen observé par chaque nœud en variant la taille de l'overlay. Pour LCC, le surcoût moyen est plus faible que celui de "Flat MeshTree" et ZigZag, et comparable à OMNI. Nous observons que pour chaque nœud LCC, et ce pour différentes valeurs de la portée R_{max}, le surcoût est régulier pour des overlays de petites tailles. La valeur maximale atteinte pour un overlay de 512 nœuds est de 1,10 kbps pour LCC avec R_{max} = 100ms, 2 kbps pour des overlays de 8000 nœuds. Les nœuds OMNI par contre, observent en moyenne un surcoût de contrôle moindre que celui de LCC. Dans ce protocole, les opérations de gestion et d'organisation de l'arbre de transmission sont à la charge de serveurs multipoints de niveau applicatif (routeurs overlays). Ainsi, les nœuds OMNI (clients) s'échangent un nombre minimum de messages de contrôle.

Nous avons également évalué le coût de la localisation, en mesurant sur PlanetLab le surcoût moyen en trafic de contrôle (en kbps). Lors de la localisation sélective, le surcoût moyen pour chaque nœud est raisonnablement peu élevé avec 0,7 kbps

1. Le code source de LCC est disponible dans le domaine public [Soft]

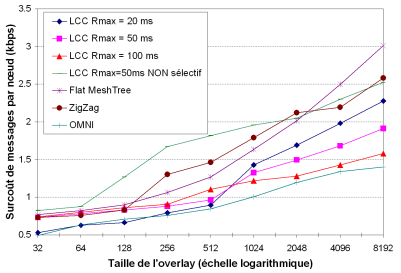

Figure 3. *Simulation des surcoûts des protocoles*

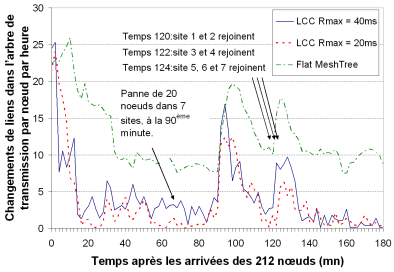

Figure 4. *PlanetLab : Fréquence d'ajustements des liens.*

pour un overlay de 212 nœuds. De plus, ce surcoût augmente faiblement en fonction du nombre de membres. Les messages de localisation sont approximativement moitié moins fréquents que ceux de la localisation non sélective. Ne pas sélectionner les nœuds augmente le surcoût dû aux opérations de mesures de distance inutiles. Cependant, nous notons que le processus de localisation sélectif pourrait nécessiter plus de temps pour localiser un nouveau venu. Nos simulations prouvent que plus de 80% des nœuds d'un overlay, composé de 2000 nœuds, sont capables de terminer le processus de localisation sélectif en contactant moins de 15 nœuds relais. En moyenne, le temps de localisation est très court avec 3,2 secondes comme valeur moyenne, un maximum de 7,2 secondes et un minimum de 1,8 secondes. Finalement, 98,4% des nouveaux venus sont capables de se connecter au plus proche groupe, et 99,3% des nœuds se connectent, 300 secondes après leur arrivée, au plus proche nœud.

4.4. *Fréquence d'ajustement des liens*

La figure 4 montre la stabilité de la structure LCC pendant des changements d'adhésions au groupe multipoint. Sur PlanetLab, nous commençons par dénombrer les ajustements de liens après que le dernier nœud s'est joint à l'overlay. Les résultats sont collectés à 30 secondes d'intervalle. Nous observons que la fréquence en ajustements descend en dessous de 5 par nœud par heure pour l'overlay LCC. Elle se stabilise par contre à environ 10 par nœud par heure pour la structure "Flat MeshTree". Pour confirmer la robustesse de LCC en cas de changements fréquents d'adhésion ou de scénarios de crashs, nous injectons des pannes simultanées de 20 nœuds dans 7 sites différents, à la 90^{eme} minute et nous les laissons rejoindre l'overlay à la 120^{eme} minute. Nous observons que l'activité des ajustements de liens dans LCC est modérée (toujours inférieure à 5 par nœud par heure) durant ces changements d'adhésion. Après la 140^{eme} minute, le taux moyen d'ajustement de lien se stabilise à 2 par nœud par heure. A cause des connexions initiales aléatoires, ce même taux pour "MeshTree" est maintenu à 10. Ceci conforte notre intuition que les schémas ne se basant pas sur une localisation initiale pourraient nécessiter un taux élevé de messages de contrôle pour les opérations de maintenance et réparation des structures.

Les résultats de simulation confirment les conclusions des expérimentations PlanetLab. "Flat MeshTree" souffre d'un taux d'ajustements des liens élevé, avec plus de 20 changements de liens par nœud par pas de simulation. Comparé à ZigZag, LCC a un taux d'ajustements très faible. Les nœuds LCC (R_{max} = 20 ms) ont une moyenne de 11,4 par nœud par pas avec un minimum de 1,4. Les changements de liens dans OMNI sont moins fréquents que les autres structures. Les nœuds OMNI exécutent en moyenne 12,9 ajustements par pas de simulation, avec un minimum de 0,78. Nos ex-

périmentations montrent aussi que dans LCC, $ARDP$ décroît rapidement à une valeur inférieure à 2 après seulement 400 secondes, i.e. moins de 14 périodes de raffinements, alors que le temps de convergence des autres protocoles est voisin de 1400 secondes.

4.5. *Pénalité en délai et stress sur le lien*

La figure 5 caractérise le délai moyen induit observé par les nœuds dans un large overlay, en observant les variations de $ARDP$ en fonction de la taille de l'overlay.

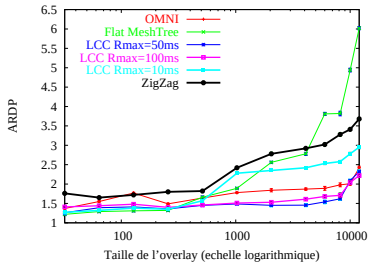

Figure 5. *Comparaison de $ARDP$ en fonction de la taille d'overlay.*

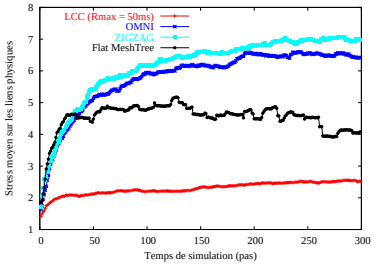

Figure 6. *Simulations : stress sur les liens.*

Dans "Flat MeshTree", $ARDP$ atteint une valeur supérieure à 4 démontrant que ce protocole ne passe pas à l'échelle de plus de quelques centaines de nœuds. Néanmoins, notons que "Flat MeshTree" a un $ARDP$ inférieur à LCC pour les groupes de petites tailles (moins de 500 nœuds). ZigZag maintient une valeur $ARDP$ stable, mais est relativement peu performant, avec un $ARDP$ égal à 2,5 dans des overlays de 3000 nœuds. Nous observons que pour LCC, $ARDP$ est inférieur de 60% à ZigZag. Pour R_{max} = 50 ms et 100 ms, les valeurs de $ARDP$ sont maintenues dans l'intervalle 1,2 et 1,4 pour de grands overlays. OMNI a aussi une valeur $ARDP$ constante (1,82) et est plus performant en moyenne que LCC pour des overlays de 12000 nœuds. Notons aussi que dans de grands overlays et pour des groupes définis par des portées de 10 ms, $ARDP$ croît pour atteindre 3, en raison d'une plus grande probabilité de dispersion des nœuds. Des portées plus grandes sont plus efficaces dans ce cas.

La figure 6 montre la variation du stress sur les liens physiques, pour les protocoles OMNI, ZigZag, "Flat MeshTree" et LCC ($R_{max} = 50ms$). Les résultats sont collectés 2000 secondes après que le dernier nœud ait rejoint l'overlay. Le stress de OMNI et ZigZag se stabilisent entre 6,5 et 7. L'overlay MeshTree obtient un stress inférieur à ces derniers avec des valeurs oscillant fortement entre 4 et 5, dues à l'effet aléatoires des connections établies. Notons enfin que LCC a un stress impressionnant entre 2,5 et 2,8, soit 2 à 3 fois moins que les autres protocoles. L'information de topologie physique est d'une importance capitale dans cette observation. En effet, les paquets sont seulement émis à travers la hiérarchie supérieure et envoyés aux dirigeants de groupes, ou potentiellement vers les nœuds ponts. Cela permet de réduire le nombre de flux redondants entrant dans chaque réseau, considérant chaque groupe comme un "réseau local", et les dirigeants comme des "routeurs multipoints".

Nous avons aussi évalué LCC par rapport aux pannes des dirigeants de groupes. Les simulations prouvent une grande adaptabilité à ces pannes (recouvrement des pannes de 30% des dirigeants en moins de 2 secondes) grâce au mécanisme pro-actif d'élections des dirigeants et des nœuds ponts connectés à la topologie supérieure.

5. Conclusion

Dans cet article, nous avons proposé une solution pratique pour améliorer l'efficacité des overlays multipoints pour de grands groupes. La construction est initiée par un processus de localisation simple, précis et passant à l'échelle qui permet à un nouveau venu de rapidement localiser les groupes de nœuds les plus proches dans la topologie physique. En se basant sur cette localisation, nous avons proposé une construction overlay hiérarchisée prenant en compte la topologie physique. Nous avons introduit des mécanismes pour s'adapter aux pannes des dirigeants de groupes ou aux changements des caractéristiques de la topologie sous-jacente. Nos expérimentations PlanetLab ainsi que nos simulations prouvent que LCC induit un surcoût peu élevé. Comparé à des approches à base de raffinements et tenant compte de la topologie, la structure LCC converge plus rapidement avec moins de fréquence d'ajustements de liens. En conclusion, LCC est très adapté à des applications à grande échelle comme des diffusions d'événements pour des milliers de participants. Dans nos travaux futurs, nous nous intéresserons à des moyens pour ajuster automatiquement différents paramètres de LCC à travers des tests Internet réels. Nous étudierons également des techniques pour sécuriser l'overlay et prévenir la fraude de certains utilisateurs.

6. Bibliographie

[BA02] S. BANERJEE, ET AL., *Scalable Application Layer Multicast*, ACM SIGCOMM, Pittsburgh, 2002.

[BA03] S. BANERJEE, ET AL., *Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications*, IEEE Infocom, San Francisco, 2003.

[CH00] Y. H. CHU, S. G. RAO, ET H. ZHANG, *A case for end system multicast*, ACM SIGMETRICS, Santa Clara, 2000.

[HE02] D. A. HELDER ET S. JAMIN, *End-host multicast communication using switch-trees protocols*, GP2PC, Berlin, 2002.

[KA06] M. A. KAAFAR, T. TURLETTI, ET W. DABBOUS, *A Locating-First Approach for scalable overlay Multicast*, IEEE IWQoS 2006, New Haven, USA, 2006.

[KW02] M. KWON ET S. FAHMY, *Topology-aware overlay networks for group communication*, NOSSDAV, Miami Beach, 2002.

[LA05] L. LAO, ET AL., *TOMA : A Viable Solution for Large-Scale Multicast Service Support*, IFIP Networking, Waterloo Ontario, 2005.

[LI03] Z. LI ET P. MOHAPATRA, *Hostcast : A new overlay multicast protocol*, IEEE ICC, Anchorage (Alaska), 2003.

[ME01] A. MEDINA, ET AL., *BRITE : Universal topology generation from a user's perspective*, Rapp. Tech TR-2001-003, Boston, 2001.

[PL] `http://www.planetLab.org`

[RA02] S. RATNASAMY, ET AL., *Topologically-Aware Overlay Construction and Server Selection*, IEEE Infocom, New York, 2002.

[SA02] S. SAROIU, P. K. GUMMADI, ET S. D. GRIBBLE, *A measurement study of peer-to-peer file sharing systems*, MMCN, San Jose (California), 2002.

[Soft] `http://www-sop.inria.fr/planete/Software/`

[SO04] J. K. SOLLINS, *Exploiting Autonomous System Information in Structured Peer-to-Peer Networks*, IEEE ICCCN, Chicago, 2004.

[TA05] S. W. TAN, A. G. WATERS, ET J. CRAWFORD, *Meshtree : A Delay optimised Overlay Multicast Tree Building Protocol*, Rapp. Tech. 5-05, U. of Kent, 2005.

[TR03] D. TRAN, K. HUA, ET T. DO, *Zigzag : An Efficient Peer-to-Peer Scheme for Media Streaming*, IEEE Infocom, San Francisco, 2003.

[WO05] B. WONG, A. SLIVKINS ET E. G. SIRER, *A Lightweight Approach to Network Positioning without Virtual Coordinates*, ACM SIGCOMM, Philadelphia, 2005.