Répartition des états de routage multipoint par les plus courts chemins multiples

Mickaël Hoerdt, Jean-Jacques Pansiot, Damien Magoni, Dominique Grad

Equipe Réseaux et Protocoles, LSIIT - Pôle API Boulevard Sébastien Brant 67400 ILLKIRCH FRANCE

Les communications de groupe nécessitent la construction d'arbres de diffusion dans les réseaux. En général, un arbre de diffusion correspond à un groupe, ainsi il se pose un problème d'extensibilité en terme de nombre de groupes gérables dans un réseau de type Internet. En pratique cela se traduit par des états de commutation à gérer sur les routeurs de l'Internet, ceux-ci étant difficilement agrégeables en raison de la propriété logique des groupes de communications. Nous proposons ici une analyse de la répartition des états de commutation en tirant partie des plus courts chemins multiples de l'Internet.

Keywords: multipoint, plus courts chemin multiples, extensibilité, états de commutation.

1 Introduction

Le problème du routage multipoint a été étudié de longue date et pourtant à ce jour aucun déploiement à grande échelle n'a eu lieu dans l'Internet : la majorité des flux de communications reste point à point alors que les travaux de recherche commencés par Deering [Dee89] datent maintenant de près de 15 ans et semblaient très prometteurs au vu des premières applications démontrées [CD92]. Aujourd'hui, l'une des raisons du non déploiement du routage multipoint est son manque d'extensibilité sur l'Internet. En effet la taille et la diversité de ce réseau dans des domaines aussi importants que les types de connexions ou d'applications constituent un obstacle non négligeable. On identifie plusieurs problèmes d'extensibilité dans le routage multipoint :

– par rapport au nombre de groupes.
– par rapport à dynamicité des groupes considérés.
– par rapport à la taille du réseau considéré.

On peut distinguer d'autres problèmes de déploiement du routage multipoint qui ne sont pas encore résolus dans les dernières architectures proposées à l'IETF. Ainsi le modèle SSM [HC99] appliqué à l'architecture PIM [EA98] ne résout pas encore les problèmes de sécurité ou d'économie liés aux communications de groupe. Notre étude porte sur l'explosion des états de routage multipoint induite par le nombre de groupes. Les paquets à destination d'un groupe sont acheminés vers les membres en parcourant un arbre de diffusion. Un noeud du réseau doit maintenir un état pour chaque arbre de diffusion auquel il appartient. La croissance importante du nombre de groupes, et du nombre d'états dans le réseau qu'elle induit, peut amener à saturation les noeuds du réseau.

Classiquement, les protocoles de routage multipoint utilisent les informations des plus courts chemins issues du routage point à point, pour construire les arbres de diffusion. Nous proposons une solution basée sur l'utilisation des plus courts chemins multiples dans la construction des arbres de diffusion permettant de limiter la saturation des noeuds par une meilleure répartition des états de routage multipoint.

Cet article est organisé comme suit : la section 2 présente les notions de plus courts chemins multiples et les principes de répartition des états de routage, la section 3 présente le problème d'extensibilité du routage multipoint par rapport au nombre de groupes, la section 4 donne une méthode nous ayant permis de résoudre un problème d'explosion combinatoire dans une simulation et enfin nous terminons par une présentation et analyse des résultats expérimentaux obtenus en section 5 et une conclusion sur nos travaux en section 6.

2 Plus courts chemins multiples et arbre de diffusion

Le problème du calcul de plus courts chemins multiples a été largement étudié dans les communications point à point sur l'Internet [NZ01]. Il en est de même pour les arbres des plus courts chemins. Or, actuellement l'impact des plus courts chemins multiples sur les arbres des plus courts chemins n'a pas encore été étudié dans la littérature, plus particulièrement sur la répartition des états de commutation sur l'Internet. En effet, il est clair que si il existe des plus courts chemins multiples, alors il peut exister plusieurs arbres des plus courts chemins pour un même groupe.

Plusieurs algorithmes distribués de construction de tables de routage point à point proposent déjà une gestion de plus courts chemins multiples comme par exemple RIP, IS-IS ou OSPF. Sur un routeur IP, cela se traduit par plusieurs prochains sauts vers une destination, ce qui peut être utilisé non seulement lors de la commutation de paquet unicast mais aussi lors de la construction d'un arbre de routage multipoint. En figure 1 on peut constater qu'il existe deux plus courts chemins entre le noeud 7 et le noeud 1, de même entre le noeud 6 et le noeud 1. Il existe alors plusieurs arbres des plus courts chemins possibles (4 exactement) si les feuilles de l'arbre sont $\{7,6\}$ et la racine 1. Si les noeuds 5 et 6 sont des routeurs IP, alors ils ont plusieurs prochains sauts possibles vers le noeud 1. Nous cherchons ici à savoir si lors de la construction d'un arbre de diffusion, il existe une fonction locale de choix du plus court chemin suffisamment simple pour être déployée sur l'Internet tout entier et quelle en serait l'influence sur la répartition des états dans le routage multipoint.

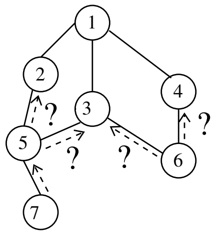


FIG. 1: Construction d'arbre avec plus courts chemin multiples

3 Extensibilité sur le nombre de groupes

Certains problèmes d'extensibilité rencontrés dans la commutation multipoint sont de même nature que ceux rencontrés dans la commutation point à point mais ne peuvent pas se résoudre de la même manière. L'un de ces problèmes, en particulier dans les réseaux à commutation de paquet de grande échelle comme l'Internet concerne l'extensibilité en terme de nombre d'états de commutation.

En effet selon le paradigme du "prochain saut" de l'Internet, chaque paquet arrivant sur un noeud de l'Internet nécessite une recherche dans la table de commutation de ce noeud, ce qui est très coûteux en ressources de calcul lorsque la taille de cette table est importante et que la diversité et le nombre de flux arrivant sur une interface sont élevés. Il existe des mécanismes de cache et des algorithmes performants de recherche dans ces tables qui permettent de réduire le temps de commutation. Ceux-ci ne sont pas suffisants à cause de la diversité des flux et du débit que nécessitent les applications de l'Internet. Ainsi l'avènement du web dans les années 93-94 a provoqué une croissance telle dans les états de commutation des routeurs qu'il a rapidement fallu déployer des mécanismes d'aggrégation tirant profit de la structure hiérarchique de l'Internet [FLYV93] sous peine de voir s'effondrer le réseau.

Pour ce qui concerne les états des communications de groupes, le problème se complique, car celles-ci sont beaucoup plus liées aux applications selon le modèle des communications de groupe défini par Deering [Dee89]. Ainsi ces états ne sont non seulement plus agrégeables car les membres d'un groupe n'ont pas de contrainte topologique mais ils peuvent être potentiellement beaucoup plus dynamiques car ils ne connectent les membres d'un groupe de taille variable que pendant la durée de l'application de groupe contrairement aux états de commutation point à point qui eux connectent les membres d'un réseau en permanence.

3.1 Combinaisons d'arbres de diffusion

Lors de l'adhésion des différents membres à un groupe, on réalise la construction d'un arbre de diffusion. Plusieurs groupes peuvent coexister au sein d'un réseau, et donc plusieurs arbres. Mais si l'on considère qu'il existe différentes façons de construire un arbre de diffusion pour un groupe donné, il existe aussi plusieurs arbres possibles par groupe. Plus précisément, étant donnés un ensemble de groupes $G_1, G_2, \ldots, G_n$ et $A_1^{G_i}, A_2^{G_i}, \ldots A_k^{G_i}$ l'ensemble des arbres des plus courts chemins pour un groupe donné G_i nous considérons alors une combinaison d'arbres comme un ensemble $A_1^{G_1}, A_2^{G_2}, \ldots, A_n^{G_n}$, c'est à dire un seul choix d'arbre par groupe en prenant tous les groupes. Ainsi selon les combinaisons considérées, les états de commutation peuvent se répartir différemment sur l'Internet.

Le problème d'extensibilité étudié ici se pose en terme du nombre d'états de commutation qu'un noeud de l'Internet pourra supporter, c'est à dire le nombre d'arbres de communication qui passent par un même noeud lorsqu'un nombre élevé de groupes existe sur le réseau. Plus précisément, on s'intéresse à la répartition de ce nombre d'états dans l'Internet et on cherche à l'améliorer [WK00], en diminuant de manière conséquente le nombre d'états sur les noeuds surchargés. Diverses propositions ont déjà été faites pour diminuer le nombre d'états de commutation [TN98, BB99]. Nous proposons ici une évaluation minorante par rapport au nombre d'états par noeuds sur l'ensemble des combinaisons d'arbres des plus courts chemins possibles afin de déterminer la viabilité de l'utilisation de plus courts chemins multiples indépendamment de l'algorithme de répartition des états utilisé.

On cherche à trouver l'ensemble des combinaisons possibles de tous les arbres possibles afin de savoir s'il existe un optimum intéressant à atteindre avec un algorithme de construction d'arbre efficace. L'espace de recherche étant très étendu, une recherche exhaustive n'est pas réaliste pour des raisons de temps de calcul. Par exemple, en imaginant qu'on combine 1000 groupes qui se déclinent chacun en seulement deux arbres des plus courts chemins, on a déjà 2^{1000} combinaisons. Nous sommes face à un problème d'explosion combinatoire que nous proposons de résoudre en calculant une solution partielle, c'est à dire en calculant seulement l'ensemble des noeuds qui sont toujours présents dans tous les arbres d'un groupe : nous appellerons ces noeuds **obligatoires** et nous détaillerons leur calcul en section 4.1.

4 Calcul d'une borne inférieure sur le nombre d'états

On se place dans le cadre des simulations sur des grands graphes (plusieurs milliers de noeuds) ayant des propriétés similaires à celle du graphe de l'Internet. Une étude similaire sur le nombre de groupes présents par noeuds a été proposée dans [WK00] mais effectuait une combinaison aléatoire à chaque mesure. Nous proposons ici une méthode permettant d'éviter de rechercher dans l'ensemble des solutions en donnant une borne inférieure en termes de nombre d'états dans toutes les combinaisons possibles (pour un même ensemble de groupes). L'obtention de cette borne inférieure nous permet d'évaluer dans quelle mesure un algorithme utilisant les plus courts chemins multiples pour la répartitions des états peut être utile dans l'Internet d'aujourd'hui.

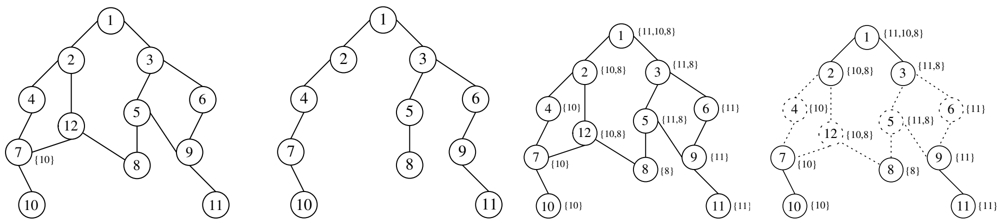
4.1 Noeuds obligatoires dans un arbre de diffusion des plus courts chemins

Soit un graphe $G = (V, E)$ avec V l'ensemble des noeuds de ce graphe et E l'ensemble des arêtes. La figure 2(a) représente un tel graphe avec $card(V) = 12$ et $card(E) = 14$.

Soit un arbre des plus courts chemins de s vers R : On sélectionne parmi V la racine qu'on appellera s, et un ensemble R de feuilles $\{r1, r2, \ldots rn\}$ puis on relie chaque élément r de R par un plus court chemin de r à s (ou de s à r car par hypothèse le graphe est non orienté pour l'instant) de telle manière à ce que l'union de l'ensemble de ces plus courts chemins soit un arbre. Un tel arbre peut être représenté par une liste de plus courts chemins depuis chaque feuille r jusqu'à s dont l'union deux à deux est toujours un arbre (deux chemins ne peuvent s'intersecter qu'une seule fois). La figure 2(b) représente un arbre possible en prenant comme plus courts chemins $\{10, 7, 4, 2, 1\}$,$\{8, 5, 3, 1\}$ et $\{11, 9, 6, 3, 1\}$.

On se propose maintenant de calculer pour une racine s et un ensemble R de feuilles l'invariant de l'ensemble des arbres, c'est à dire l'ensemble T des noeuds présents dans l'ensemble des arbres des plus courts chemins possibles. De manière triviale, s et R sont inclus dans T, et pour calculer les autres noeuds, on procède de la façon suivante :

Pour chaque feuille r de R, on calcule l'ensemble des plus courts chemins de s à r et on étiquette les noeuds de ces plus courts chemins par r. On a donc une liste de plus courts chemins pour chaque feuille. La figure 2(c) représente un tel étiquetage avec $s = 1$ et $R = \{10, 8, 11\}$. Pour $r = 10$, les chemins sont $\{10, 7, 12, 2, 1\}$ et $\{10, 7, 4, 2, 1\}$, pour $r = 8$ on a $\{8, 5, 3, 1\}$ et $\{8, 12, 2, 1\}$, pour $r = 11$ on a les chemins $\{11, 9, 6, 3, 1\}$ et $\{11, 9, 5, 3, 1\}$.



(a) Graphe exemple

(b) Exemple d'arbre des plus courts chemins

(c) Marquage du graphe

(d) Noeuds obligatoires et optionnels d'un arbre des plus courts chemins

FIG. 2: Différents ensembles d'un même graphe

Le calcul de T se fait suivant la hauteur h d'un noeud donné. De manière triviale la racine est à une hauteur 0, pour calculer tous les noeuds de l'ensemble des arbres potentiels ayant une hauteur h, on utilise la liste de liste des plus courts chemins qu'on a calculée auparavant. La table 1 donne l'alignement des plus courts chemins par hauteur pour tous les plus courts chemins calculés précédemment. Seule la racine a une hauteur nulle, les noeuds de hauteur 1 sont $\{2,3\}$, ceux de hauteur 2 sont $\{4,12,5,6\}$, ceux de hauteur 3 sont $\{7,8,9\}$ et ceux de hauteur maximum $\{10,11\}$.

h=4	h=3	h=2	h=1	h=0
10	7	4	2	1
10	7	12	2	1
	8	5	3	1
	8	12	2	1
11	9	6	3	1
11	9	5	3	1

TAB. 1: Hauteur des différents noeuds de l'exemple

Ensuite, pour calculer T, on prend l'ensemble des noeuds d'une hauteur donnée h : si un noeud est le seul qui permette d'atteindre une feuille donnée de l'arbre, alors il appartient à T. Dans ce but, on réalise une soustraction des étiquettes contenant les feuilles accessibles par un noeud à une hauteur donnée. La figure 2(d) illustre les noeuds n'étant pas dans T par des pointillés. On trouve que les noeuds $\{s,r,2,3,7,8,9\}$ sont obligatoires pour le groupe et donc seront dans tous les arbres possibles du groupe, quels que soient les plus courts chemins choisis depuis les feuilles. Les noeuds optionnels sont $\{4,12,5,6\}$.

5 Résultats expérimentaux

Une simulation a été réalisée sur un graphe respectant les propriétés de la topologie de l'Internet [MP01], les résultats ont été moyennés sur 10 graphes en créant 1000 groupes aléatoirement d'une densité variant autour de 1% sur chaque graphe, chaque graphe ayant 3000 noeuds. On calcule un pourcentage de saturation de ces noeuds, c'est à dire le nombre de groupes pour lesquels ils sont obligatoires ou optionnels par rapport au nombre de groupes total sur la topologie. La figure 3 réprésente un classement pourcentile du pourcentage de saturation pour les groupes obligatoires par noeuds et les groupes optionnels et obligatoires par noeuds. On obtient une borne maximum et minimum du pourcentage de saturation par pourcentile qu'on compare ensuite à un pourcentage de saturation obtenu en créant une combinaison d'arbres suivant diverses méthodes de calculs des plus courts chemins afin d'estimer une marge de manoeuvre réaliste dans la répartition des états liés à la construction d'un arbre de routage multipoint. La figure 4 garde le même classement pourcentile en indiquant la part de groupes pour lesquels les noeuds d'un pourcentile sont optionnels en utilisant Dijkstra pour calculer les plus courts chemins entre chaque membre du groupe et la racine. On rappelle que les noeuds ont été marqués avec l'algorithme présenté en section 4.1 avant d'exécuter Dijkstra.

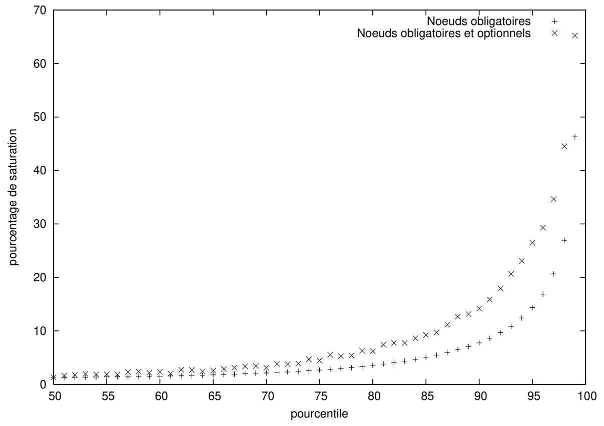


FIG. 3: Pourcentage de saturation par noeuds

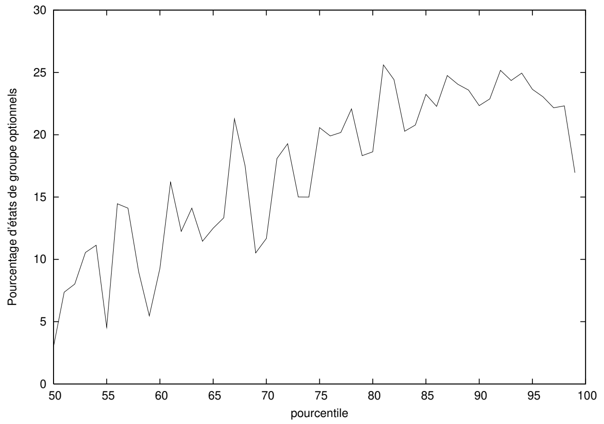


FIG. 4: Part d'états optionnels pour Dijkstra

On peut constater que la courbe de répartition du pourcentage d'états pour lesquels un noeud donné est obligatoire suit une exponentielle, ce qui indique qu'avec la topologie de l'Internet actuel, même le meilleur des algorithmes de répartition ne pourra pas empêcher que certains noeuds peu nombreux doivent supporter une majeure partie du trafic multipoint (plus de 40% dans les noeuds de plus fort pourcentile). Malgré cela, on peut remarquer en figure 4 que la part des états pour lesquels un noeud est optionnel reste intéressante dans les noeuds de pourcentile fort. Par exemple si un noeud contient 1000 états de commutation, qu'il est obligatoire pour 800 états et optionnel pour 200 états, il contient 80% d'états pour lesquels il est obligatoire, et contient une part de noeuds optionnel de 20%. Cette part peut ensuite être répartie selon diverses méthodes simples que nous proposons d'évaluer dans la section 5.1.

5.1 Exploitation des résultats

Il est clair que l'amélioration de la répartition des états sur l'Internet ne pourra se faire que sur les états de commutation optionnels sur les noeuds. Nous cherchons ici à savoir si une fonction simple sur un noeud permettrait d'améliorer cette répartition, en particulier sur les noeuds les plus saturés. Pour que cette fonction reste simple, il faut qu'elle utilise au maximum les informations locales au noeud, par exemple

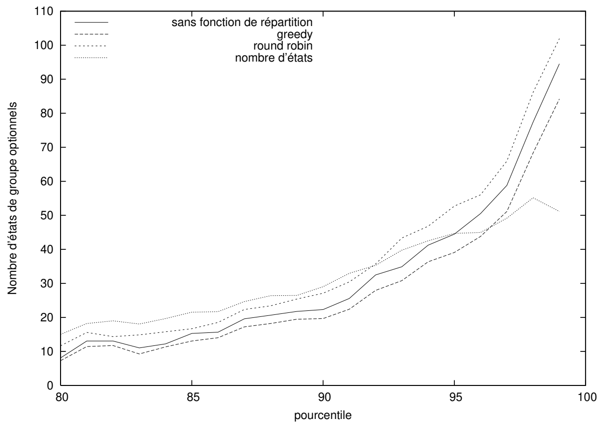


FIG. 5: Nombre d'états optionnels par fonction de répartition

le ou les prochains sauts vers une destination fournis par la table de routage point à point. Lorsqu'il existe plusieurs prochains sauts depuis un noeud vers la racine d'un arbre on peut décider de répartir selon diverses fonctions les branches créées sur les prochains noeuds en direction de cette racine. Ainsi pour un arbre en particulier on peut décider de passer par un prochain saut en particulier : on a une fonction de répartition locale au noeud qui distribue les différentes branches d'arbre créées en amont vers les sources de données multipoint. Nous proposons trois fonctions de répartitions que nous évaluons par la suite :

1. *Le prochain saut dans la construction d'une branche est celui qui contient le moins d'états.*
2. *Le prochain saut est choisi séquentiellement. (round robin)*
3. *Le prochain saut à la priorité s'il est déjà dans l'arbre.(greedy)*

En cas d'égalité dans la première fonction on prend le premier trouvé, et pour la troisième si l'arbre n'est pas trouvé on choisit le prochain saut par défaut. En figure 5 on peut voir une comparaison du nombre d'états optionnels par noeuds entre ces différentes fonctions de répartition et sans répartition. La fonction sur le nombre d'états se révèle la plus intéressante avec une diminution de moitié du nombre d'états optionnels sur les noeuds les plus saturés. Par contre la fonction round-robin n'améliore pas la répartition des états. Ces résultats sont à pondérer avec la limite théorique qu'on a obtenue en section 5. En effet les fonctions de répartition ne peuvent agir que sur les états de groupes optionnels. De plus on peut constater qu'on a réussi à décharger les noeuds les plus saturés mais que cette décharge a tendance à se répercuter sur les noeuds les moins saturés. Par exemple pour la première fonction, le nombre d'états optionnels est plus élevé pour les noeuds en dessous du 95ème pourcentile.

6 Conclusion et perpectives

Nous avons proposé une classification en deux types de noeuds dans un ensemble d'arbre des plus courts chemins pour un même groupe donné. Nous avons utilisé cette classification pour montrer qu'il est possible de décharger les noeuds contenant le plus d'états dans l'Internet en utilisant des fonctions locales de répartition très simples, ce qui est non négligeable si l'on prend en compte qu'un mécanisme complexe peut se révéler impossible à déployer sur une topologie aussi étendue et diversifiée que l'Internet. Nous avons aussi calculé une bonne inférieure irréductible d'états de commutation quels que soient les combinaisons d'arbres envisagées et donc quel que soit le mécanisme de répartition proposé. Les résultats que nous avons obtenus sont à pondérer en fonction de la taille des graphes utilisés. En effet on pourrait supposer qu'avec l'augmentation de la taille du graphe, on augmente le gain de la répartition des états, ce que nous proposons d'évaluer dans de futurs travaux.

Références

[BB99] Ljubica Blazevic and Jean-Yves Le Boudec. Distributed core multicast (dcm) : a multicast routing protocol for many groups with few receivers. In *NGC*, Pisa, Italy, November 1999.

[CD92] Stephen Casner and Stephen Deering. First ietf internet audiocast. *ACM Computer Communications Review*, 22(3), July 1992.

[Dee89] Stephen Deering. RFC 1112 : Host extensions for IP multicasting., August 1989.

[EA98] Deborah Estrin and Al. RFC 2362 : Protocol independent multicast-sparse mode (PIM-SM) : Protocol specification., June 1998.

[FLYV93] Vince Fuller, Tony Li, Jessica Yu, and Kannan Varadhan. RFC 1519 : Classless inter-domain routing (CIDR) : an address assignment and aggregation strategy., September 1993.

[HC99] Hugh W. Holbrook and David R. Cheriton. IP multicast channels : EXPRESS support for large-scale single-source applications. In *SIGCOMM*, pages 65–78, Cambridge, Massachusetts, September 1999. ACM.

[MP01] Damien Magoni and Jean-Jacques Pansiot. Influence of network topolgy on protocol simulation. In *ICN*, Colmar, France, July 2001.

[NZ01] Srihari Nelakuditi and Zhi-Li Zhang. On selection of paths for multipath routing. In *IWQoS*, Karlsruhe, Germany, June 2001.

[TN98] Jining Tian and Gerald W. Neufeld. Forwarding state reduction for sparse mode multicase communication. In *INFOCOM (2)*, pages 711–719, 1998.

[WK00] Tina Wong and Randy Katz. An analysis of multicast forwarding state scalability. *IEEE/ACM Transactions on Networking*, 8(3) :419–427, 2000.