



HAL
open science

Gestion de la Qualité de Service par les Hypercubes dans un Environnement de Routage Multipoint : Cas des Protocoles PIM-SM et MPLS

Mohamed Ech-Cherif El Kettani

► **To cite this version:**

Mohamed Ech-Cherif El Kettani. Gestion de la Qualité de Service par les Hypercubes dans un Environnement de Routage Multipoint : Cas des Protocoles PIM-SM et MPLS. CFIP'2009, Oct 2009, Strasbourg, France. inria-00419465

HAL Id: inria-00419465

<https://inria.hal.science/inria-00419465>

Submitted on 23 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Gestion de la Qualité de Service par les Hypercubes dans un Environnement de Routage Multipoint :

Cas des Protocoles PIM-SM et MPLS

Mohamed Dafir ECH-CHERIF EL KETTANI

*Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS)
Université Mohammed V-Souissi
BP 713 – Rabat Agdal – Maroc
dafir@ensias.ma*

RÉSUMÉ. Les problèmes de communication de groupe ont été résolus partiellement par la mise en place de protocoles de routage multipoint, supportant des applications de groupe. Le champ d'utilisation et le domaine d'application de tels protocoles dépend de plusieurs facteurs, dont la répartition des membres du groupe sur le réseau, le nombre de sources, et les ressources disponibles. Cependant, certains aspects du routage multipoint ne sont pas encore résolus de manière efficace, à savoir: le déploiement à grande échelle du routage multipoint en vue de sa généralisation à tout le réseau Internet, et la difficulté de mettre en œuvre une politique efficace de gestion de la Qualité de Service liée au groupe. Dans cet article, nous proposons « Hyper-LDP », une stratégie de diffusion multipoint basée sur les Hypercubes, qui associe à un groupe des garanties de Qualité de Service. Cette solution est voisine de l'approche classique dite « overlay »: en effet, aux nœuds du domaine de routage est associé tout d'abord un hypercube, à la base duquel est construit l'arbre de diffusion associé au groupe. Appliquée à un domaine MPLS, « Hyper-LDP » automatise l'affectation des labels et la recherche de chemins dans un domaine MPLS. Adaptée au protocole PIMSM, elle permet d'anticiper la construction de l'arbre de diffusion avant la couche 3, au même niveau que les labels MPLS. Nous testons la faisabilité de la solution, par simulation puis sur maquette en combinant MPLS et PIM-SM.

ABSTRACT. The problem of multicasting messages to group members is now solved thanks to many network level multicast protocols. However, their deployment is an issue that depends on many factors. Thus, existing multicast routing protocols suffer from lack of scalability, and absence of group dedicated QoS policies. The goal of this paper is to study these possibilities, by proposing "HyperLDP", a " Hypercube QoS enabled multicast strategy ", which constructs the multicast tree before layer 3, at the same level than MPLS labels. We introduce this notion, and, from a set of simulation results, we analyse the performance of this solution and apply it to PIM-SM implemented on an MPLS backbone.

MOTS-CLÉS: Routage multipoint, Hypercube, Qualité de Service, MPLS, Simulation.

KEY WORDS: Multicast routing protocols, Hypercube, QoS, MPLS, Simulation.

1. Introduction

Le routage multicast est indispensable pour déployer les applications distribuées basées sur l'algorithmique de groupe. Celles-ci engendrent des flux multimédia importants, depuis une (plusieurs) source(s) vers les membres du groupe.

Les problèmes liés à la diffusion restreinte de messages aux membres d'un groupe ont été résolus partiellement grâce aux protocoles de routage multipoint supportant diverses applications de groupe (TV sur ADSL, etc.). En pratique, ces applications seront performantes si la structure de communication du groupe (l'arbre de diffusion), est en adéquation avec le protocole de routage multipoint.

Ainsi, notre principal objectif est de proposer une stratégie de diffusion multipoint extensible, destinée à associer à un groupe des garanties de Qualité de Service. La solution se base sur les Hypercubes, et se rapproche de l'approche classique dite « overlay ». Mais elle s'en distingue, car il s'agit tout d'abord de modéliser le domaine de routage unicast dans une structure d'hypercube (cette opération est paramétrable); puis dans un second temps d'en déduire l'arbre de diffusion du groupe.

Après cette introduction, nous effectuons dans la section 2 de brefs rappels sur le routage multipoint et la QoS. La section 3 propose une méthode associant un hypercube à un domaine de routage intégrant à la fois la QoS et le routage multipoint. La section 4 présente Hyper-LDP, une approche automatisant l'affectation des labels et la recherche de chemins pour les groupes dans un domaine MPLS, par les hypercubes. Nous testons la faisabilité de la solution, à travers une implémentation basée sur les protocoles MPLS et PIM-SM. Les résultats obtenus sont analysés, en comparant la solution proposée aux stratégies existantes, pour enfin conclure.

2. Le routage multipoint par la Qualité de Service

2.1. Position du problème

Deux aspects majeurs et complémentaires du routage multipoint n'ont pas encore été résolus : d'une part, le déploiement du routage multipoint à grande échelle, et d'autre part, les garanties en matière de Qualité de Service nécessaires aux applications de groupe. Nous proposons une solution globale qui déploie le routage multipoint sur tout un domaine, en apportant au groupe des garanties de QoS.

Afin de bien délimiter le contexte de notre étude, nous allons effectuer un certain nombre d'hypothèses de départ qui justifieront les choix ultérieurs, sur deux plans:

Pour le routage multipoint, nous examinerons les différentes approches de routage existantes, afin de retenir le protocole le mieux adapté à notre étude.

Pour la gestion de la QoS, nous passerons en revue les approches existantes afin de définir l'environnement intégrant les garanties nécessaires au routage multipoint.

2.2. Intégration de la Qualité de Service au routage multipoint

2.2.1. Modèles de gestion de la QoS et MPLS

Le défi de la technologie Internet est d'utiliser des flux multimédia offrant une qualité de transmission équivalente à celle des réseaux traditionnels. Le protocole qui s'impose dans ce sens est MPLS, pour ses qualités d'extensibilité, de flexibilité des modèles de service (qualitatives, Platinum, Gold, Silver, etc) et de signalisation.

Il existe une grande analogie entre les composantes de l'architecture DiffServ et celles de MPLS: Un nœud du réseau, « Label Switching Router » (LSR), possèdera à la fois une table de routage et de commutation. On distingue trois types de LSR :

- *Ingress Edge LSR (routeur d'entrée)* : il classe les paquets IP reçus dans des FEC à qui sont associées des étiquettes ou labels. Il encapsule ces paquets dans des unités de données MPLS étiquetées, et les commutent en sortie.
- *Interior LSR (routeur interne)* : son rôle consiste à recevoir des unités de données MPLS étiquetées, et commutent ces PDU en fonction de leur étiquette.
- *Egress Edge LSR (routeur de sortie)* : il reçoit des unités de données MPLS étiquetées, supprime l'en-tête MPLS des PDU reçues, et route les paquets IP.

Chaque équipement met en œuvre certains mécanismes : le conditionnement, l'établissement, et l'acheminement des trafics. Il fait appel à diverses ressources telles que « FEC » pour la classification des flux, « LSP » pour la détermination du chemin idéal, et « LDP » pour la distribution de labels. Ceci évite, à l'inverse de RSVP, de surcharger les processeurs des routeurs traversés par ces types de flux.

2.2.2. Insuffisances du Traffic Engineering appliqué au multicast sous MPLS

Le mécanisme d'ingénierie de trafic (TE) offre au routage IP une grande flexibilité grâce au routage explicite (bande passante, QoS,...). Le chemin MPLS (LSP) n'est plus déterminé à chaque bond (routage implicite) : l'ingress LSR choisit le chemin de bout en bout, alors que l'egress LSR se limite à l'analyse du label MPLS. Il existe diverses méthodes de remplissage des tables d'acheminement:

- La première est manuelle: cette solution n'est réaliste que pour un nombre très limité de classes d'équivalence FEC (Forwarding Equivalence Class).
- La seconde consiste à construire automatiquement et de proche en proche les chemins MPLS (LSP) pour chacune des classes d'équivalence reconnues dans les tables de routage, sur la base des informations dans les tables de routage IP.
- Un **contrôle centralisé** peut établir des LSP et fournir la QoS à assurer. Il s'appuie sur deux protocoles: RSVP-TE (une variante de RSVP), ou CRLDP [ASH 02] (une extension de LDP dont il requiert la présence). CR-LDP est une alternative à RSVP-TE, implémentée par TCP, interagissant avec LDP, et dont les fonctions sont réalisées par instructions ASIC.
- M-LDP [WIJ 09] est une approche récente qui n'a pas encore été normalisée. Du point de vue de la gestion de la QoS, ces mécanismes n'incluent pas le routage multipoint.

2.3. Généralisation du routage multipoint au réseau Internet

2.3.1. Difficulté de servir le multicast dans un domaine MPLS

Le déploiement du multicast dans un domaine MPLS reste une opération délicate:

Tout d'abord, la construction du LSP nécessiterait outre les LSPs point-à-point, d'établir les LSPs point-à-multipoint, et multipoint-à-multipoint.

De plus, comme le routage multipoint utilise l'adresse IP multicast, l'agrégation de trafic risque d'être délicate si les récepteurs d'un même groupe étaient dispersés.

La construction d'un arbre de diffusion suppose que tous les nœuds du domaine de routage interviennent dans ce processus, alors que MPLS accorde une préférence aux routeurs de bordure (Ingress LSR) pour déterminer le chemin de bout en bout.

Enfin, comme MPLS s'insère entre les couches 2 et 3, la construction de l'arbre de diffusion d'un groupe (couche 3) obligerait la remontée des divers traitements depuis la couche « liaison de données », vers la couche « réseau » dans les LSRs. Ceci obligerait à mixer routage et commutation au niveau des opérations de gestion

2.3.2. Limitations des protocoles de routage multipoint sur MPLS

De nouveaux protocoles intègrent dorénavant le routage multipoint à MPLS:

PIM-MPLS utilise les messages d'adhésion de PIM-SM pour distribuer les labels MPLS et construire l'arbre, sans intégrer le TE, ne considérant ni la charge du réseau ni le routage optimal.

MMT [BOU 05] construit un arbre dans un réseau MPLS en considérant seulement les routeurs de branchement de cet arbre. Malgré ses performances, MMT n'offre pas de mécanismes de gestion de la QoS associée à un groupe.

2.3.3. Cas du protocole PIM-SM

Notre choix a porté sur le protocole PIM-SM, car cette étude vise le déploiement du multicast à grande échelle, ce qui ne peut être envisageable pour DVMRP ou MOSPF.

PIM-SM, dans sa version 2 [FEN 06] souffre de certaines limitations:

- La multitude de sources favorise les goulots d'étranglement sur l'arbre partagé unique. Une alternative est PIM-SSM [HOL00] qui considère une seule source.
- La gestion manuelle des points de RendezVous (RP) limite les possibilités de réadapter selon des critères, tels que la répartition des récepteurs, etc.
- La transition entre arbre partagé et arbre par source ne se fait pas selon des critères explicitement définis, ce qui rend cette condition très subjective.

Néanmoins, PIM-SM se distingue par les avantages suivants:

- Le processus de diffusion est indépendant du nombre de sources du groupe. La stratégie d'arbre partagé garantit un nombre d'états par routeur en $O(G)$.
- Les Points de RendezVous (RP) réduisent le coût global de l'arbre de diffusion.

PIM-SM intègre en plus depuis la version 2, les fonctionnalités de PIM-DM. Suite à cette analyse, le travail présenté considérera le protocole PIM-SMv2 en vue de généraliser le routage multicast à l'ensemble du réseau Internet.

3. La communication de groupe par les Hypercubes

Un protocole de routage multicast fonctionne selon la classe d'arbre de diffusion associée, qui se base soit sur l'optimisation des ressources du graphe, soit sur la minimisation du chemin entre source(s) et récepteurs. Nous distinguons 3 classes d'arbres : par source, partagé, et hybride.

3.1. Propriétés des Hypercubes

Par définition, un Hypercube de dimension N peut être défini en considérant ses $n=2^N$ sommets comme étiquetés par les représentations binaires des nombres de 0 à 2^N-1 . Deux sommets sont connectés si leurs étiquettes diffèrent par un bit exactement. Le degré et le diamètre de l'hypercube sont égaux à N , qui possède un nombre d'arêtes $S_N = N \times 2^{N-1}$. L'hypercube possède une complexité en nombre d'états égale à $n \times \log(n)$, n étant égal au nombre de sommets.

La manipulation des hypercubes dans un graphe passe par les phases suivantes:

- Construction de l'hypercube
- Numérotation des arêtes de l'hypercube
- Calcul du plus court chemin entre 2 sommets de l'hypercube

3.2. Algorithme de communication de groupe dans un Hypercube

L'acheminement des paquets dans un hypercube dépend de la nature de la communication:

3.2.1. Communication unicast dans un hypercube

Plusieurs algorithmes de routage unicast existent. Nous utiliserons à titre indicatif « l'algorithme du e-cube » (figure 1), pour sa simplicité, en déterminant le choix de l'ordre de parcours (par numéro de bit décroissant). Cette stratégie est exempte d'inter blocage. L'algorithme emprunte un des plus courts chemins.

Ainsi, pour un nombre « i » de bits différents entre source et destination, il existe $i!$ plus courts chemins possibles. Pour se rendre du nœud « 00000 » au « 11101 », $i=4$. L'hypercube résiste à la contention (modification du plus court chemin de 2 messages de la même source vers la même destination) et aux pannes.

```

Procédure envoyerUnicast_eCube(message m, message cfg, adresseSource n1, adresseDestination n2)
Début
  Stocker "cfg" dans la mémoire du nœud courant // utilisé ultérieurement pour configurer le nœud
  Calculer  $i = \text{distance}(n1, n2)$  indiquant le nombre de bits différents entre "n1" et "n2"
  Si ( $i \neq 0$ ) alors
    Localiser les "i" bits qui diffèrent entre "n1" et "n2"
    Envoyer "m" au voisin "n" de "n1" ayant en commun le 1er bit différent de "n2"
    envoyerUnicast_eCube(m, n, n2)
  Sinon
    voisin = destination // Ne rien faire, la destination est atteinte
Fin

```

Figure 1. Communication unicast (eCube)

3.3. Consommation en ressources

Quelles sont les performances de cet algorithme, et la complexité associée en terme de ressources utilisées? Soit H^N un hypercube de dimension N .

- L'hypercube dispose d'un nombre $n=2^N$ sommets.
- Le nombre d'arêtes noté « S_N » est $S_N = N \times 2^{N-1} = \ln(n) \times n / (2 \times \ln(2))$.
- Le calcul du plus court chemin entre 2 sommets de l'hypercube est en $O(N)=O(\log(n))$, par définition d'un hypercube de dimension N .

Au niveau du nombre d'états mémorisés, chaque nœud va posséder une combinaison des états suivants:

- une entrée relative à son numéro de sommet sur chaque nœud de l'arbre: elle comporte 1 état (occupant N bits)
- une entrée de type $(*, G)$ sur chaque nœud de l'arbre de diffusion associé au groupe G . Cette entrée comportera la liste des nœuds voisins appartenant à l'arbre, qui comporte au maximum N états.
- une entrée de type (R, G) sur la source unique du groupe, contenant la liste des récepteurs du groupe, en vue de sa consolidation lors de la phase 2. Cette entrée unique comportera au maximum 2^N états.

Si on se limite à la phase 1, le nombre d'états par nœud de l'arbre associé à cet algorithme est en $O(G \times N)=O(G \times \log(n))$, G étant le nombre total de groupes.

Si on inclut également la phase 2 (qui reste optionnelle), le nombre d'états pour la source augmente. En effet, il faut inclure les entrées de type (R,G) . La complexité associée au nombre d'états de la source est donc en $O(G \times N + G \times 2^N)=O(G \times n + G \times \log(n))=O(G \times n)$, G étant le nombre total de groupes.

- Comparaison avec la complexité de PIM-SM:

Si on considère uniquement la phase 1, la complexité est meilleure que celle déployée dans PIM-SM, qui est en $O(G \times N)$.

Si on considère également la phase 2, la complexité change uniquement pour la source, qui est en $O(G \times n)$.

3.4. Application au protocole PIM-SM

Comme l'hypercube constitue la structure de communication du réseau, il s'agira d'établir une correspondance entre l'arbre de diffusion convenu par PIM-SM et les parties de l'hypercube impliquées dans la diffusion dans le groupe.

L'approche adoptée par PIM-SM [FEN 06], est initiée par les récepteurs. Le point de rendez-vous RP (« Rendez-vous Point ») est le point de rencontre entre les sources et les récepteurs d'un groupe. Appliqué à PIM-SM, l'algorithme précédent doit subir donc une légère adaptation qui n'apportera aucune modification au niveau de la complexité. En effet, on remplacera la source du groupe par le point de RendezVous (RP). Au lieu que l'arbre de diffusion ait pour racine la source du groupe, c'est le point de RendezVous qui jouera ce rôle. Ainsi, la(les) source(s) enverra(enverront) à son(leur) tour le trafic de groupe au RP qui s'occupera de compléter la communication.

4. Hyper-LDP: Gestion de la Qualité de Service par les Hypercubes

4.1. Caractéristiques générales de la solution

4.1.1. Modélisation d'un domaine MPLS par un hypercube

Nous supposons dans un premier temps que le domaine MPLS correspond à un réseau composé de 2^N LSR, et qu'il a une structure régulière d'hypercube. Nous traiterons ensuite le cas général d'un réseau quelconque, qui en sera une adaptation.

Dans le cas d'un hypercube régulier, chaque LSR dispose de N voisins auxquels il est relié: il dispose d'une adresse qui tient sur N bits (voir figure suivante):

Interface In	Label In	Destination	Interface Out	Label Out
le0	1100	127.0.0.1	le0	1100

Figure 4. Représentation d'un nœud dans la table NHFLE

L'identificateur du nœud sera stocké à titre indicatif dans la table NHFLE, sous un format particulier. On pourra penser par exemple à une entrée ayant comme valeur d'interface (In et/ou Out) « le0 » (adresse de boucle locale), et comme « label in » et « label out » l'identificateur du nœud dans l'hypercube (figure 4).

Cette modélisation facilitera l'affectation des labels d'entrée et/ou de sortie pour chaque nœud LSR. En effet, comme un label a une taille de 20 bits, nous réserverons N bits pour stocker l'identificateur du nœud. Les 20-N autres bits permettront d'affecter au nœud des valeurs qui éviteront d'une part d'entrer en conflit avec les valeurs de labels réservées, et de stocker éventuellement des informations utiles : identifier la longueur d'onde pour GMPLS, ou l'appartenance du nœud à l'arbre de diffusion d'un groupe (voir section 3.2.2). Ce choix est volontaire, de sorte à établir une relation entre l'identificateur du nœud et le label.

Étant donné que le label est utilisé pour identifier un trafic appartenant à différentes FEC, la distribution des labels sera en relation étroite avec les valeurs des identificateurs de nœuds dans l'hypercube. Cette opération offre l'avantage de construire une table de commutation de labels dans chaque nœud, qui déduira ses valeurs automatiquement à partir des identificateurs des nœuds voisins dans l'hypercube. Les valeurs stockées dans cette table permettront de passer par le plus court chemin, par analogie avec le fonctionnement du routage dans un hypercube.

La commutation nécessite la connaissance préalable des informations suivantes:

- le numéro du LSR courant,
- le numéro du LSR voisin qui correspond au prochain saut
- le numéro du LSR final, qui correspond au Egress-LSR

Comme les LSR sont identifiés par leur numéro dans l'hypercube, nous pouvons intégrer les 3 informations précédentes dans 2 champs d'une entrée de NHLFE:

D'une part, le « Label In » sera constitué de 2 parties: une partie gauche indiquant le nœud courant, et une partie droite indiquant le nœud destination.

D'autre part, le « Label Out » sera à son tour constitué de 2 parties: une partie gauche indiquant le prochain saut, et une partie droite indiquant la destination.

Dans l'exemple de la figure 5, on associera au réseau destination 12.15.4.0/24 le noeud final (Egress-LSR) « 1111 ». Le LSR courant est identifié par la valeur « 1010 », et le noeud voisin situé sur le chemin vers la destination est identifié par la valeur « 1001 ». On associera donc à cette entrée de la table les labels:

- Label In=« 10001111 », résultat de la concaténation de « 1000 » et « 1111 »
- Label Out=« 11001111 », résultat de la concaténation de « 1100 » et « 1111 »

Interface In	Label In	Destination	Interface Out	Label Out
s0	10001111	12.15.4.0/24	s1	11001111

Figure 5. Format d'une entrée de la table NHFLE

Ainsi, l'identification du prochain saut devient simple à réaliser: Après avoir extrait le numéro du noeud destination à partir des labels, le choix du prochain saut sera réalisé en comparant le numéro du noeud final, et le numéro du noeud courant. La correspondance avec l'interface de sortie est alors déduite de l'entrée de la table NHFLE (s1 dans notre cas). La figure suivante est une illustration:

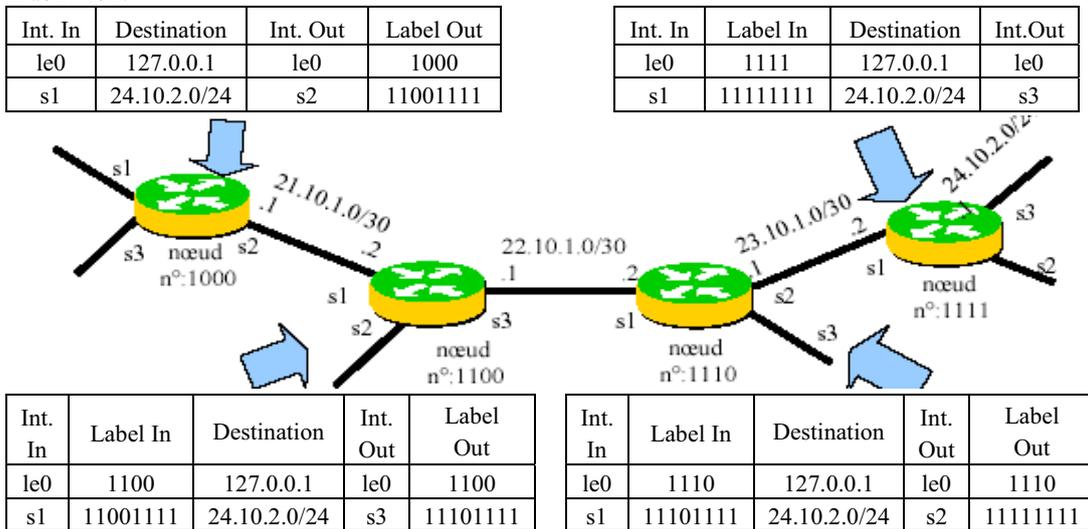


Figure 6. Acheminement d'un message depuis le noeud '1000' vers le réseau 24.10.2.0/24

4.1.2. Avantages de la proposition

Sur un réseau modélisé sous forme d'hypercube de dimension N, la proposition du paragraphe précédent présente les avantages suivants:

- En terme de commutation, la distance d'acheminement des données à destination avec un nombre de sauts égal à N.
- Dans notre cas, MPLS ne sera pas obligé d'apprendre la topologie du réseau en consultant les tables de routage se basant sur des protocoles tels que OSPF, etc.
- Ceci n'est pas sans conséquences sur l'ingénierie de trafic: le remplissage des entrées de tables NHFLE sera effectué de manière quasi-automatique.

Cette solution favorisera l'implémentation du partage et équilibrage de charge.

4.2. Opérations de gestion d'un domaine MPLS

4.2.1. Adaptation des Hypercubes à un domaine MPLS

Dans la section 4.1, nous avons supposé que le réseau d'interconnexion avait une structure régulière d'hypercube, ce qui est loin de la réalité: Il est indispensable de revoir donc les hypothèses de départ, de sorte à « hypercubiser » le réseau.

La première opération consistera donc à sélectionner les routeurs MPLS que l'on souhaite impliquer dans l'opération (routeurs ayant une forte capacité, et dispersés à travers le domaine MPLS): ces routeurs joueront le rôle de jonction entre les différentes parties du réseau. Ce nombre est paramétrable, c'est à dire que l'administrateur peut choisir un nombre 'n' quelconque, qu'il pourra modifier par la suite, selon les contraintes du réseau. On pourra démarrer avec $n=2$, puis 4, 8, 16...

On désignera un routeur central, responsable de l'opération «d'hypercubisation» du domaine, ainsi que des routeurs de relève, pour la tolérance aux pannes.

La construction de l'hypercube est réalisée selon une approche ascendante, en construisant des hypercubes de dimension 1, puis en les regroupant afin de produire des hypercubes de niveau 2, etc. Le choix des associations entre nœuds sera effectué en calculant et en comparant les distances réelles reliant les différents nœuds. La métrique n'est pas figée (nombre de sauts, délai, etc.)

```

Procédure génèreHypercube(int n, noeud[] r={r0,..., rm}, noeud[] t={n0,n1,...,ni,...,n2N-1})
// n : taille de l'hypercube, r : le réseau global, t : ensemble des sommets de l'hypercube
Début
  Si (n < 2N) alors
    Sélectionner les (2N-n) nœuds qui seront comptabilisés en double
    // On choisira les nœuds les plus performants, et on considèrera que n = 2N
    Pour i = 1 à n faire
      Pour j = 1 à (2N+1-i-1) faire
        Calculer les 2N-i distances minimales de chacun des (i-1) -Cubes, noté Ci-1,j
        Calculer distances complémentaires reliant les (i-1)-Cubes obtenus Ci-1,j selon les dist. minimales calculées
        Construire les 2N-i(i)-Cubes
      FinPour
    FinPour
  Fin

```

Figure 7. Algorithme de construction d'un hypercube sur un domaine MPLS

4.2.2. Performances de la solution proposée

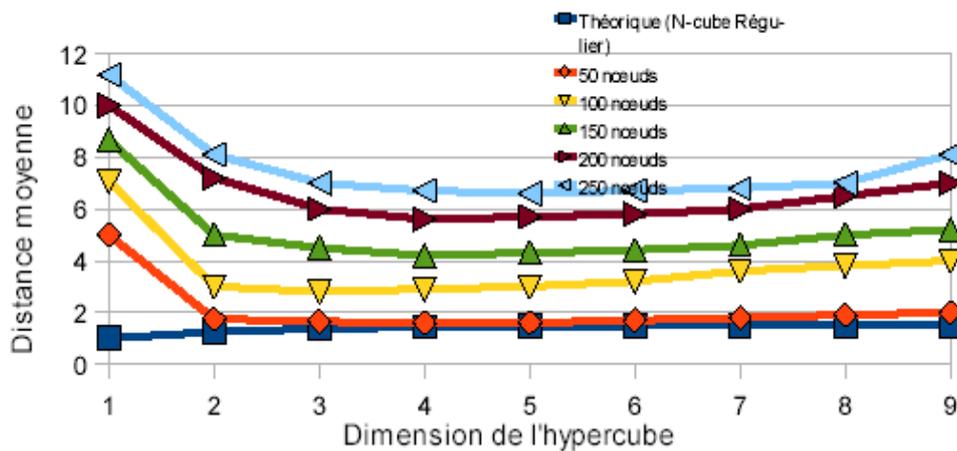
Lors de l'opération de construction de l'hypercube, le problème classique des chemins redondants risque de se poser. C'est à dire que pour atteindre une destination sur l'hypercube, on risque d'effectuer un aller retour sur un même lien, à cause de la numérotation en vigueur.

Ce problème peut être résolu simplement, puisque chaque nœud dispose dans sa table NHFLE les diverses entrées qui mènent vers la destination finale. Il suffira donc de vérifier lorsqu'un paquet arrive sur un routeur situé entre 2 nœuds virtuellement adjacents de l'hypercube, si la destination ne figure pas sur le nœud intermédiaire, avant d'arriver au prochain saut de l'hypercube.

Quelques simulations (30) ont été effectuées sur des réseaux dont la taille varie entre 16 et 128 nœuds, en vue de comparer les performances d'un hypercube issu de l'algorithme ci-dessus, avec celles d'un hypercube régulier. La métrique considérée est le nombre de sauts. Pour un hypercube de dimension N ayant $n=2^N$ nœuds, la distance moyenne séparant un nœud des autres sommets est:

$$d_N = (2d_{N-1} + 1) / 2^N$$

Figure 8: Distance moyenne d'un nœud de l'hypercube



Les simulations montrent que l'algorithme donne des résultats qui se rapprochent des résultats théoriques pour un hypercube régulier.

4.2.3. Hyper-LDP: une politique de gestion de la QoS

La distribution des labels sera effectuée conformément à l'algorithme précédent pour tous les nœuds, sauf pour les nœuds redondants. En effet, comme les nœuds redondants correspondent à des nœuds logiques fusionnés dans un seul nœud physique du réseau, nous conserverons cette distinction logique, en supposant que le nœud physique dispose d'autant d'entrées locales (de type « le0 », « le1 », etc.), avec des labels ayant des valeurs significatives, conformément à la proposition précédente. Si l'acheminement d'un paquet nécessite de passer par plusieurs nœuds logiques se trouvant sur le même nœud physique, alors le paquet effectuera les transitions nécessaires localement jusqu'à trouver le chemin qui le fera acheminer vers un voisin. Ceci ne pénalisera pas le temps globale de traitement du paquet.

De plus, la solution proposée permet dorénavant de définir une nouvelle approche de distribution des labels, qui ne tient pas compte des protocoles de routage sous-jacents (OSPF, etc.), tel que c'est le cas actuellement. Il s'agit donc d'une approche offre des avantages par rapport à CR-LDP ou RSVP-TE.

Étant donné que l'algorithme proposé suit une approche ascendante, il est supporte naturellement le mécanisme de « stack », connu dans MPLS.

Cette solution aide enfin l'opérateur à faire évoluer l'opérateur vers une structure de communication, se rapprochant d'un hypercube, ce qui n'est pas sans avantages.

4.3. Application de Hyper-LDP au routage multicast

Nous avons expérimenté la stratégie proposée dans les sections 3.2 et 4.1, en utilisant une maquette implémentant les protocoles MPLS pour la gestion de la QoS, et PIM-SM pour la gestion du routage multipoint. Nous avons utilisé les implémentations sur Linux, qui sont: MPLSLinux1.962, et pimd_2.1.0-alpha29.

Le domaine MPLS comporte le 10 routeurs LSR, dont 4 sont des LSR de bordure. Un point de RendezVous est placé associé à un LSR du domaine MPLS. Le logiciel de routage Pimd a été installé sur chacun des routeurs. Tous les LSRs de bordure sont configurés comme candidat RP pour une liste de groupes Multicast.

L'évaluation de la plateforme mise en place a été effectuée par des mesures de bande passante (avec IPERF) et de latence (avec SMOKEPING). Pour ce faire, le lecteur VLC est utilisé pour générer des flux multimédia unicast et multicast.

En terme de QoS, nous avons appliqué une CoS dédiée au flux multicast, en choisissant AF3 1 pour le champs DSCP, et 3 pour EXP. Le reste du trafic unicast fonctionne en mode best-effort. La maquette fonctionne selon une distribution des labels conforme à la stratégie proposée.

5. Conclusion

Cette étude a montré qu'il est possible d'envisager une nouvelle vision pour la gestion de la QoS dans un domaine de routage MPLS, basée sur les hypercubes. Nous avons proposé une approche offrant beaucoup d'avantages, dans la mesure où la gestion de la QoS tire profit de la topologie du réseau, s'il est correctement pensé. De plus, cette approche peut donner lieu à une nouvelle politique de gestion de la QoS, qui serait une alternative à CR-LDP et RSVP-TE, qui dépendent fortement des protocoles de routage, tels que OSPF. L'application de cette proposition au cas particulier du routage multicast ne peut être que bénéfique. Nous avons effectué des tests relatifs à la faisabilité, et ils s'avèrent satisfaisants.

Ce travail peut être étendu de manière à automatiser dorénavant la distribution des labels selon l'approche préconisée, et des tests à plus grande échelle, afin de tester les performances de la solution avec des paramètres supplémentaires tels que la latence et l'encombrement du réseau.

Remerciements

L'auteur tient à remercier l'opérateur « Ittissalat Al-Maghrib » IAM pour son soutien financier dans le cadre du projet de R&D dans le cadre du contrat N° 10510010279.06.PI

Bibliographie

- [ASH 02] J. ASH, M. GIRISH, E. GRAY, B. JAMOSSI, G. WRIGHT, « Applicability Statement for CR-LDP », *RFC 3213*, Network Working Group, January 2002
- [WIJ 09] I. WIJNANDS ET AL., « In-band signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths », *Internet-Draft*, NWG, 2009
- [BOU 05] A. BOUDANI, « MMT Routage multicast gestion des petits groupes et ingénierie de trafic », *thèse de Doctorat*, 2005
- [FEN 06] FENNER, ET AL., « Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised) », *RFC 4601*, August 2006.
- [HOL 00] H. HOLBROOK, B. CAIN, « Source-Specific Multicast for IP », PIM-SSM, *Internet Draft*, IETF, <draft-holbrook-ssm-arch-00.txt>, July 2000.