

Études et simulations de schémas de routage compact : cas du schéma d'Abraham

Frédéric Majorczyk

LaBRI

Projet DCR Alcatel-Lucent

04-12-2009

Problématique

- Routage au niveau des AS (Autonomous System)
- réseau d'une taille de ≈ 30000 nœuds en accroissement
- Schéma de routage inter AS utilisée : BGP

Limites de BGP

- Augmentation de la taille des tables de routage
- Non compensée par les augmentations de performance/capacité liées au matériel ni par l'ajout de mécanismes d'optimisation
- taux de croissance des tables évalué à 1.2/1.3

Routage compact et simulation

Routage Compact

- Schémas de routage visant à réduire la taille des tables de routage de chaque nœud
- Augmentation de l'étirement ou stretch
- Comment les évaluer ?

Simulation

- Simulateurs performants : nombre de messages échangés, taille des tables et taille du réseau
- Granularité de la simulation

DRMSim

DRMSim

- Dynamic Routing Model Simulator
- Simulateur à événements discrets
- Développé à Sophia-Antipolis par Luc Hogié dans le cadre du projet DCR

Caractéristiques

- Conçu pour simuler de grands réseaux
- Nombreuses topologies de réseau disponibles (grilles, complet, GLP, etc.)
- Aucune limitation sur les algorithmes pouvant être implémentés

Présentation du schéma

- Compact Name-Independent Routing with Minimum Stretch (Abraham et al., 2008)
- Optimal
 - Étirement maximal de 3
 - Tables locales en $\tilde{O}(\sqrt{n})$
- Combinaison de techniques “standard” en routage compact
- Repose en partie sur Routing in Trees (Fraigniaud-Gavoille, 2001)

Routing in Trees

- Schéma de routage dans les arbres utilisant $O(\log(n))$ bits pour les adresses et la mémoire locale
- Soit un arbre T dont la racine est r
- T_u est le sous-arbre de T dont la racine est u
- Poids de u : $w(u)$ est le nombre de nœuds dans T_u
- DFS un peu spécial : le premier fils visité est le fils le plus "lourd"
 - définit un id pour chaque nœud
- rang d'un nœud u , fils de v : $rank(v, u)$ nombre d'enfants de v qui ont une étiquette inférieure à $id(u)$

Routing in Trees (2)

- $port(u, v) = 1$ si v est le père de u , sinon $1 + rank(u, v)$
- $path(u) = (rank(r, u_1), \dots, rank(u_{t-1}, u))$ où $r, u_1, \dots, u_{t-1}, u$ est le plus court chemin de r à u
- $cpath(u)$: suite des rangs du chemin de la racine r à u où l'on a enlevé tous les 1

- Adresse de u : $l(u) = \langle id(u), cpath(u) \rangle$
- Mémoire en u : $table(u) = \langle id(u), w(u), w_1(u), |cpath(u)| \rangle$
- $ROUTE(x, l(y)) = \begin{cases} 0 & \text{si } id(x) = id(y) \\ 1 & \text{si } id(y) < id(x) \text{ ou } id(y) \geq id(x) + w(x) \\ 2 & \text{si } id(x) < id(y) \leq id(x) + w_1(x) \\ p + 1 & \text{sinon avec } p = (|cpath(x)| + 1)^{me} \text{ élément de } cpath(y) \end{cases}$

Boules de voisinage et coloriage

Boules de voisinage

- Boule de voisinage de u : $B_k(u)$ u et ses k plus proches voisins
- Propriété : si $v \in B_k(u)$ et w est sur un plus court chemin de u à v alors $v \in B_k(w)$
- Dans le cas du schéma de routage : $k = 4 * \sqrt{n} * \log(n)$

Coloriage

- Partition en \sqrt{n} ensembles (color-sets) vérifiant 2 propriétés :
 - Au plus $2 * \sqrt{n}$ nœuds dans un color-set
 - Chaque nœud a au moins un B-voisin de chaque color-set
- $c(u)$: couleur du nœud u
- Fonction de hashage h qui transforme un nom en couleur

Landmark et arbre partiel de plus court chemins

Landmark

- Choix d'une couleur comme la couleur "landmark"
- Pour un nœud v , on note l_v le nœud landmark le plus proche de v

Arbre partiel de plus court chemins

- $T(u)$: arbre de plus court chemin dont la racine est u
- $\mu(T(u), v)$: informations de routage stockées en v pour le routage dans l'arbre $T(u)$ suivant "Routing in Trees"
- $\lambda(T(u), v)$: adresse de v dans l'arbre $T(u)$ suivant "Routing in Trees"
- Chaque nœud v stocke $\mu(T(u), v)$ ssi $u \in B(v)$

Informations de Routage

B-Voisinage

- Chaque nœud stocke son B-voisinage et l'arête pour atteindre chaque nœud du B-voisinage
- u stocke le nom w et le port $u \rightarrow y$ pour $w \in B(u)$
- \Rightarrow chaque nœud sait router vers ses B-voisins suivant un plus court chemin

Landmarks

- Pour chaque nœud landmark l , le nœud u stocke $\mu(T(l), u)$ et $\lambda(T(l), l)$
- \Rightarrow chaque nœud sait router vers les landmarks suivant un plus court chemin

Informations de Routage (2)

Arbre partiel de plus court chemin

- Chaque nœud u stocke $\mu(T(x), u)$ ssi $x \in B(u)$

Couleur et hashage

- Si $c(u) = h(v)$, le nœud u va stocker des informations sur un chemin parmi les deux choix suivants :
 - $\langle \lambda(T(l_v), l_v), \lambda(T(l_v), v) \rangle$
 - \Rightarrow le message passe par l_v puis arrive à v suivant deux plus courts chemins
- Un chemin plus complexe décrit sur le transparent suivant

Informations de Routage (3)

Couleur et hashage (2)

- On considère un chemin $P(u, w, v)$ avec les propriétés suivantes :
 - Plus court chemin de u à w et de w à v
 - $u \in B(w)$
 - Il existe une arête (x, y) sur le plus court chemin de w à v
 - $x \in B(w)$ et $y \in B(v)$
- $\langle \lambda(T(u), w), x, (x \rightarrow y), \lambda(T(y), v) \rangle$

Routage de u à v

$$c(u) = h(v)$$

- Premier cas (par l_v) :
 - le nœud u ajoute à l'entête du message :
 $\langle \lambda(T(l_v), l_v), \lambda(T(l_v), v) \rangle$ et envoie le message suivant
l'algorithme de routage de "Routing in Trees"
- Second cas (par w) :
 - Routage suivant l'arbre $T(u)$ jusque w (possible grâce aux arbres partiel de routage)
 - Routage de w à x puis y par le B-voisinage et le lien $x \rightarrow y$
 - Routage de y à v suivant l'arbre $T(y)$ (possible pour la même raison)

Autre cas

- Le nœud u envoie le message à $w \in B(u)$ tel que
 $c(w) = h(v)$ et w peut envoyer le message à v

Généralités

Version "centralisée"

- Les tables sont construites avant un quelconque envoi de message de manière centralisée
- La définition des messages échangés pour construire les tables fait partie des futurs travaux
- La version actuelle permet d'obtenir des résultats sur l'étirement

Performances

- La construction des tables est coûteuse : calcul de plus court chemins et d'arbres de plus court chemins
- Borne à 500 nœuds
- Optimisations encore nécessaires

Quelques notions d'implémentation

- 5 tables différentes (codée sous forme d'HashMap, à optimiser)
 - Ajout de la couleur du B-voisin dans la table de voisinage
 - Ajout du nœud y dans la seconde table répondant à la propriété $c(u) = h(v)$
- 3 entêtes différents
 - Routage vers l_v
 - Routage suivant $P(u, v, w)$ avec ajout dans l'entête d'un champ Step indiquant dans quelle partie du routage se trouve le message
 - Routage vers un B-voisin ayant la couleur du hashé : entête contenant le nom du B-voisin
- Modification à l'algorithme : court-circuit lorsque la destination est dans le B-voisinage

Protocole des expérimentations

Protocole

- Routage de tous les nœuds vers tous les autres
- Différentes topologies : grid, GLP
- 200 nœuds (la simulation dure 1min30 en moyenne)
- Pour les topologies aléatoires (type GLP) : moyenne sur les expérimentations (valeur souhaitée 30)

Résultats

- Étirement multiplicatif et additif
- Moyenne, maximum

Exemple de sortie

8 metric(s) found:

- NumberOfRoutingFailures (1 value(s))

value of interest: 0.0

- NumberOfDeliveries (101 value(s))

value of interest: 100.0

- NumberOfRouters (1 value(s))

value of interest: 200.0

- NumberOfLinks (1 value(s))

value of interest: 790.0

- AvgStretch (100 value(s))

value of interest: 1.0033333333333332

- AdditiveStretch (100 value(s))

value of interest: 1.0

- AvgAdditiveStretch (100 value(s))

value of interest: 0.01

- Stretch (100 value(s))

value of interest: 1.3333333333333333

Résultats

GLP

- Moyenne sur 5 exécutions
- Étirement maximal : 3
- Étirement moyen : 1.16242
- Étirement additif maximal : 4
- Étirement additif moyen : .46244

Grid

- 2 exécutions (6 min de simulation par exécution)
- Étirement maximal : 3
- Étirement moyen : 1.09470
- Étirement additif maximal : 24
- Étirement additif moyen : 1.19613

Dynamicité

- Disparitions temporaires ou permanentes de liens ou de nœuds
- Mesure du temps de convergence des tables de routage
- Question ouverte !
- Que faire si un landmark n'est plus valide ?
- proposition : Calcul de plusieurs "Abraham"

Politique de routage

- Le routage des messages entre les AS est également régi par les relations entre les AS
- Plusieurs relations entre AS sont possibles :
 - Customer to provider ($C \rightarrow P$)
 - Provider to customer ($P \rightarrow C$)
 - Peering ($Pe \rightarrow Pe$)
- Ces relations peuvent être modélisées par des étiquettes sur les arêtes
- Un chemin valide pour un message est composé d'une séquence d'arêtes de ce type :
($C \rightarrow P$), \dots , ($C \rightarrow P$), ($Pe \rightarrow Pe$), ($P \rightarrow C$), \dots , ($P \rightarrow C$)

Politique de routage (2)

- Ces relations sont gérées par BGP par différents mécanismes (redondance dans l'AS path, absence d'annonce de certaines routes, etc.)
- Comment intégrer ces relations dans le schéma d'Abraham ?

Merci

Questions ?