DHT Basée sur la Géométrie Hyperbolique

Cyril Cassagnes, Telesphore Tiendrebeogo, Yérom-David Bromberg, Damien Magoni

DHT Basée sur la Géométrie Hyperbolique

Cyril Cassagnes — Telesphore Tiendrebeogo — David Bromberg — Damien Magoni

Université de Bordeaux – LaBRI – UMR CNRS 5800
351 cours de la libération, F-33400 Talence Cedex, France
{cassagne,tiendreb,bromberg,magoni}@labri.fr

RÉSUMÉ. *Dans ce papier, nous proposons un système de réseau recouvrant pair-à-pair structuré libre de toutes topologies et implantant une couche de routage par clef. Le but est de construire une couche de routage pour une DHT extensible et fiable afin de créer et de gérer de nouvelles applications dans Internet. Nous proposons une approche utilisant la géométrie hyperbolique.*

ABSTRACT. *In this paper, we propose a structured peer-to-peer overlay system with a free topology and embedding a key based routing layer. The aim is to build a routing layer for a scalable and reliable DHT in order to create and managing new applications over the Internet. We propose an approach using the hyperbolic geometry.*

MOTS-CLÉS : *Routage Glouton, Pair-à-Pair, DHT*

KEY WORDS: *Greedy Routing, Peer-To-Peer, DHT*

1. Introduction

De manière générale, une architecture pair-à-pair (P2P) structurée basée sur le concept de *Table de Hachage Distribuée* (DHT) peut être divisée en trois couches : l'application utilisant la DHT, la table de hachage distribuée elle-même et le routage par clef. Les solutions existantes tels que Chord, CAN, Pastry, Tapestry, Viceroy, Kademlia, Koorde, etc, appelées architectures P2P structurées, sont nombreuses [LUA 05]. Dans notre cas, chaque pair à un nom qui est un identifiant au niveau applicatif lequel permet aux membres de se contacter. C'est pourquoi, les correspondances nom-adresse des pairs doivent être stockées afin de lier l'espace de nommage des pairs (considéré comme l'espace des clefs) et l'espace d'adressage (considéré comme l'espace des adresses). Les adresses des pairs sont des coordonnées virtuelles prises dans le plan hyperbolique. En effet, la transmission des paquets dans le réseau recouvrant requiert un schéma d'adressage et de routage dédié. Avec ces éléments, nous pouvons construire une couche de routage par clef et ainsi créer un lien entre le nom et l'adresse d'un nœud membre afin de fournir une flexibilité des communications, comme expliqué dans des travaux antérieurs [CAS 10]. Le papier est organisé comme suit. Nous expliquons comment créer un arbre d'adressage hyperbolique dans un réseau recouvrant et comment les paquets sont routés dans un tel réseau en section 2. La section 3 décrit les mécanismes de stockage et de résolution de la DHT. La section 4 présente nos résultats de simulations avant de conclure.

2. Arbre hyperbolique

Le modèle que nous utilisons pour représenter le plan hyperbolique est appelé le modèle du disque de Poincaré. Dans ce modèle, représenté par un disque de rayon 1, les points ont des coordonnées complexes et un module complexe inférieur à 1 [KLE 07, BEA 06].

Création de l'arbre d'adressage : Les coordonnées hyperboliques (i.e. nombre complexe) des pairs sont utilisées comme les adresses dans le réseau recouvrant. Le degré q de l'arbre d'adressage est fixé par le premier pair au commencement et pour toute la durée de vie de l'arbre. Le premier pair prend l'adresse hyperbolique $(0; 0)$ et devient la racine de l'arbre. La racine peut assigner q adresses alors que les autres peuvent assigner $q-1$ adresses. Un exemple d'arbre hyperbolique avec $q = 3$ est montré sur la Figure 1. Fixer le degré de l'arbre d'adressage à q n'empêche pas le réseau recouvrant de s'étendre car un pair peut se connecter à n'importe quels autres pairs du réseau recouvrant à chaque instant. En revenche, Un pair calcule ses coordonnées dans le plan hyperbolique en fonction d'un seul de ses voisins qui devient son père dans l'arbre d'adressage. En effet, la connaissance globale du réseau n'est pas nécessaire et permet à l'algorithme d'être scalable. Alors que la méthode de [KLE 07] nécessite un algorithme centralisé sur la globalité du réseau pour trouver le plus haut degré. L'algorithme 1 montre comment les adresses sont calculées par les nœuds. Toutes les étapes de l'algorithme présentés sont adaptées pour un calcul distribué et asynchrone.

Algorithme 1 : Calculer les coordonnées des descendants d'un nœud

```
CalcFilsCoords(pair, q) ;
begin
    pas ← argcosh(1/sin(π/q));
    angle ← 2π/q;
    FilsCoords ← pair.Coords;
    for i ← 1, q do
        FilsCoords.rotationGauche(angle);
        FilsCoords.translation(step);
        FilsCoords.rotationDroite(π);
        if FilsCoords ≠ pair.ParentCoords
        then
            StockFilsCoords(FilsCoords);
        endif
    endfor
end
```

Algorithme 2 : Router un paquet dans le réseau recouvrant

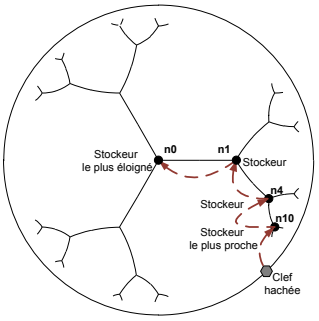
```
ProchainSaut(pair, paquet) return Pair ;
begin
    w = paquet.destinationPairCoords;
    m = pair.Coords;
    d_min = argcosh(1 + 2 |m−w|² / ((1−|m|²)(1−|w|²)));
    p_min = pair;
    forall the voisin ∈ pair.voisin do
        n = voisin.Coords;
        d = argcosh(1 + 2 |n−w|² / ((1−|n|²)(1−|w|²)));
        if d < d_min then
            d_min = d;
            p_min = voisin;
        endif
    endfall
    return p_min
end
```

Routage dans le réseau recouvrant : Un nouveau pair peut commencer à émettre des paquets dans le réseau recouvrant après avoir obtenu une adresse d'un pair de l'arbre d'adressage. Au niveau applicatif, les pairs communiquent avec leur nom. Le mécanisme de résolution nom-adresse utilisé par les pairs est la DHT présenté en section 3. Le processus de routage est effectué dans chaque pair traversé depuis l'émetteur jusqu'à la destination en utilisant un algorithme glouton basé sur les distances hyperboliques. À la réception d'un paquet, un pair calcule la distance entre la destination et chacun de ses voisins puis il transfère le paquet au voisin dont la distance est minimale, voir l'algorithme 2. Si aucun voisin n'est plus proche que le pair lui-même alors le paquet a atteint un minimum local. Dans ce cas, d'autres méthodes, non décrites dans cet article, doivent être utilisées pour réussir à router le paquet vers la destination [CVE 09].

3. Nommage et liaison dans le plan hyperbolique

Dans cette section, nous expliquons comment notre réseau recouvrant stocke et retrouve des informations dans les pairs du réseau. L'algorithme de routage hyperbolique est celui présenté en section 2. Au démarrage, chaque nouveau membre du réseau recouvrant choisit un nom qui identifie l'équipement hôte. Ce nom est conservé par l'équipement aussi longtemps que le réseau recouvrant existe. Ensuite, le pair tente de se connecter à d'autres pairs du réseau recouvrant. Enfin, l'équipement obtient une adresse hyperbolique en trouvant un père dans l'arbre d'adressage. À chaque pair est associé un couple (nom, adresse). Ce couple permet de créer le lien entre le nom et l'adresse (en anglais « *binding* »). Si le nom du pair est déjà une clef de la DHT alors un message d'erreur est envoyé en réponse au pair afin qu'il choisisse un autre nom ainsi la structure de la DHT elle-même garantit que les noms sont uniques. Un couple est stocké par son créateur à interval de temps régulier sinon il est supprimé. Un message de suppression peut aussi être envoyé par le créateur pour supprimer le couple.

Pour stocker le couple (nom, adresse) du pair dans le système, le pair hache son nom avec l'algorithme SHA-1. La clef hachée de 160 bits est divisée arbitrairement en 5 sous-clefs de 32 bits. Le pair sélectionne alors la première sous-clef et calcule un angle par une transformation linéaire. L'angle est donné par l'équation 1. Cet angle nous donne un point virtuel sur le cercle (la *clef hachée*). La Figure 1 montre comment et où un couple est stocké dans le réseau recouvrant. Ensuite, le pair détermine les coordonnées hyperboliques du pair le plus proche de ce point virtuel $n10$; Le pair $n10$ sera un stockeur du couple (nom, adresse). Pour stocker dans le pair $n10$, une requête est envoyée et routée en utilisant le routage hyperbolique de la section 2. Si la requête échoue parce que le stockeur ne peut être atteint, à cause d'un nœud ou d'un lien défectueux, alors la requête est redirigée vers le stockeur le plus proche lequel est le père du stockeur calculé $n4$ Figure 1. Ce processus continue tant que la requête n'a pas atteint un stockeur valide. La requête peut donc remonter dans l'arbre d'adressage vers la racine laquelle est le stockeur le plus éloigné. Néanmoins, l'algorithme garantit que les pairs sont stockés en priorité dans le stockeur le plus proche du cercle unité.



$$\alpha = 2\pi \times \frac{\texttt{sous-clef 32bits}}{\texttt{0xFFFFFFFF}} \qquad (1)$$

Le pair calcule alors un point virtuel v sur le disque unité en utilisant l'angle :

$$v(x,y) \text{ avec } \begin{cases} x = cos(\alpha) \\ y = sin(\alpha) \end{cases} \qquad (2)$$

Figure 1. *Système DHT hyperbolique.*

La profondeur d'un pair dans l'arbre d'adressage est définie comme le nombre de pairs le séparant de la racine (racine incluse). La valeur maximale de la profondeur des stockeurs dans l'arbre d'adressage est fixée à la configuration. Pour la redondance, nous avons plusieurs possibilités. En

effet, le nombre des sous-clefs et le nombre des copies sont des paramètres qui peuvent être fixés à la création du réseau recouvrant. La division de la clef de hachage en 5 sous-clefs est arbitraire et nous permet d'avoir au maximum une redondance de 5. Cependant nous pourrions augmenter ou réduire le nombre de sous-clef. Pour les copies, nous pouvons utiliser deux mécanismes de redondances :

1) Soit utiliser des sous-clefs distribuées uniformément et les stockeurs qui leurs correspondent.

2) Soit stocker les couples dans plusieurs ancêtres du stockeur (chemin dans l'arbre).

Néanmoins, dans le cas d'un arbre d'adressage déséquilibré, plusieurs pairs pourraient être stockés proche de la racine et donc surcharger les nœuds. Pour remédier à ce problème, les stockeurs ne pourront contenir qu'un nombre limité de couple, les requêtes de stockages supplémentaires seront refusées et redirigées vers un autre stockeur. Une augmentation du nombre de copie mène à un compromis entre fiabilité et perte de l'espace de stockage mais ces mécanismes permettent à la DHT de faire face à une croissance non uniforme du réseau recouvrant. Donc, comme dans beaucoup de systèmes existants, les couples seront stockés avec une stratégie hybride. Pour finir, notre DHT est consistante, si un stockeur échoue seul ses clefs sont perdues, les autres stockeurs ne sont pas impactés et l'ensemble du système reste cohérent.

4. Simulations

Dans cette section, nous présentons les résultats des simulations que nous avons menées pour évaluer l'utilisabilité de notre système de liaison et de notre couche de routage basé sur les coordonnées du plan hyperbolique. Nous utilisons un simulateur réseau à évènement discret de niveau paquet appelé *nem* [MAG 05] pour l'obtention de tous les résultats de simulations présentés.

Configurations et paramètres : Pour évaluer notre DHT sur une topologie réelle, nous utilisons une carte IPv6 avec 4k nœuds de 2004. La carte provient de données réelles collectées avec *nec* [1] et CAIDA [2]. Le simulateur gère un temps de simulation et chaque pair du réseau recouvrant démarre à un temps donné pour une durée donnée sur un nœud de la carte tiré au hasard. Le pair qui crée le réseau recouvrant reste actif pour toute la durée d'une simulation. Chaque pair a une durée de vie aléatoire fixée selon une probabilité décrivant une loi exponentielle avec $\lambda = 10^{-5}$ laquelle donne une valeur médiane de 300 secondes et une valeur au 90ème percentile de 1000 secondes. Le nombre des nouveaux pairs est fixé à 30 par minute avec un temps d'inter-arrivée aléatoire fixé selon une probabilité décrivant une distribution exponentielle. Les pairs créent des liens dans le réseau recouvrant avec d'autres pairs en sélectionnant ceux qui sont le plus proche en terme de saut réseau. Comme chacune des simulations dure 1 heure, cette distribution des durées de sessions des pairs produit de nombreux mouvements d'apparitions et de disparitions des nœuds, appelés remous. Du fait des remous, les nœuds forment un réseau recouvrant qui peut avoir une topologie différente de la carte et donc des longueurs de chemin moins optimales. Cela reste vraie même lorsque les pairs essaient d'établir des liens judicieux avec des pairs plus proches en terme de saut.

1. http ://www.labri.fr/perso/magoni/nec
2. Université de Californie, San Diego - The Cooperative Association for Internet Data Analysis : http ://www.caida.org/home

Sur les graphiques suivants chaque point est la moyenne de 20 exécutions et les valeurs d'écart-type associées sont représentées par une barre d'erreur. Nous considérons qu'à un instant donné seulement quelques nœuds sont actifs comme pairs du réseau recouvrant. La fréquence des requêtes générées dans chaque pair est de 1 toutes les 30 secondes pour le stockage et de 1 toutes les 5 secondes pour la résolution. Les requêtes sont délivrées en tenant compte du temps de transmission des liens et nous collectons les données toutes les 600 secondes. Nous observons l'influence du degré de l'arbre d'adressage sur les performances des requêtes de stockages et de résolutions. Plus précisement, nous mesurons le taux de réussite ainsi que la longueur moyenne des chemins dans le réseau recouvrant pour les deux types de requêtes. En revanche, nous n'étudions pas les stratégies de redondance expliquées en section 3.

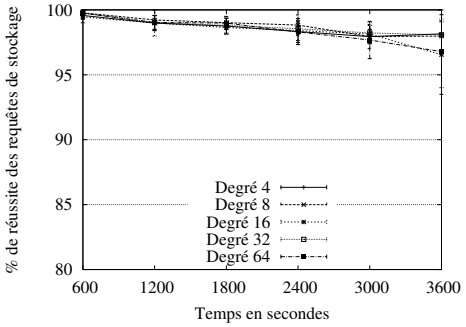


Figure 2. *Pourcentage de réussite des requêtes de stockages*

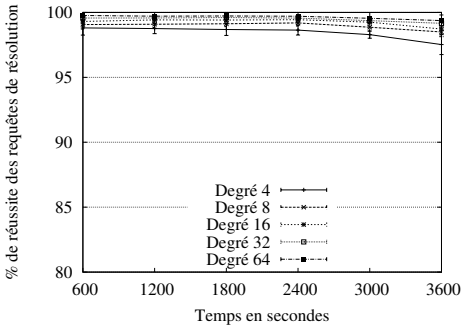


Figure 3. *Pourcentage de réussite des requêtes de résolutions*

Les Figures 2 et 3 montrent le taux de réussite des requêtes au cours du temps. Nous observons que le taux de réussite est élevé malgré les remous. Précisons que seulement une copie du couple de liaison est stockée dans le système et rappelons qu'une liaison est une association entre le nom et l'adresse d'un pair où le nom agit comme une clef.

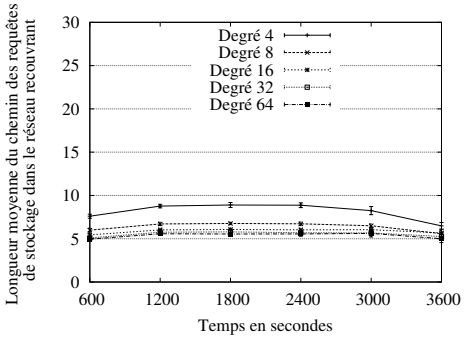


Figure 4. *Distance moyenne des requêtes de stockages dans le réseau recouvrant*

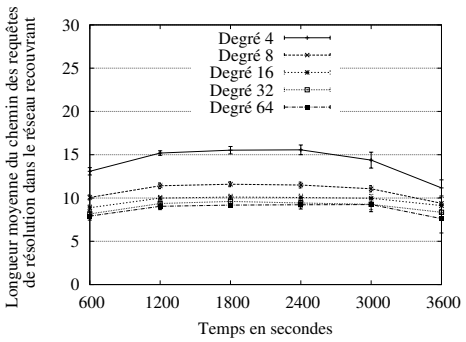


Figure 5. *Distance moyenne des requêtes de résolutions dans le réseau recouvrant*

Les Figures 4 et 5 montrent la longueur moyenne des chemins des requêtes dans les nœuds du réseau recouvrant. Le nombre des pairs à traverser, destination incluse, avant le stockage varie entre 6 et 9 en fonction du dégré q de l'arbre d'adressage. Cette valeur diminue quand le degré augmente mais se stabilise à chaque fois. Pour les requêtes de résolutions, le nombre des pairs à traverser, en incluant le chemin retour, varie de 9 à 16. Donc, le nombre de sauts reste cohérent. De plus, les courbes ont la même tendance, donc le système reste stable malgré les remous. Le taux de réussite des différentes requêtes est encourageant pour un réseau recouvrant s'exécutant 1 heure avec un total de 1800 pairs. Les longueurs moyennes des chemins pour les requêtes sont proches des valeurs obtenues avec ce type de système.

5. Conclusion

Dans ce papier, nous avons proposé un réseau recouvrant pair-à-pair basé sur la géométrie hyperbolique qui fournit à ses membres une infrastructure de routage fiable ainsi qu'une infrastructure de liaison. Le plan hyperbolique utilisé au travers du modèle du disque de Poincaré est parfaitement adapté pour fournir des coordonnées virtuelles lesquelles nous permettent d'avoir à la fois un routage hyperbolique et une structure de DHT consistante comme expliqué en section 3. Nos résultats de simulations ont démontré pour des réseaux dynamiques comprenant jusqu'à 4k pairs que le taux de réussite des requêtes de stockages et de résolutions est encourageant et cela même en présence de nombreux remous. Toutefois, dans des travaux futurs, nous prévoyons l'étude de scénarios plus complexes avec des cartes bien plus grandes, nécessitant du trafic de donnée entre les pairs. Mesurer le débit entre des pairs est une tâche ardue à modéliser et nécessiterait soit une simulation plus détaillée, soit l'utilisation d'un prototype réel.

6. Bibliographie

[BEA 06] BEARDON A. F., MINDA D., « The hyperbolic metric and geometric function theory », *International Workshop on Quasiconformal Mappings And Their Applications*, 2006, p. 9-57.

[CAS 10] CASSAGNES C., BROMBERG D., MAGONI D., « An Overlay Architecture for Achieving Total Flexibility in Internet Communications », *International Conference on AITM*, 2010.

[CVE 09] CVETKOVSKI A., CROVELLA M., « Hyperbolic Embedding and Routing for Dynamic Graphs », *proceedings of the 28th IEEE International Conference on Computer Communications*, April 2009.

[KLE 07] KLEINBERG R., « Geographic Routing Using Hyperbolic Space », *proceedings of the 26th IEEE International Conference on Computer Communications*, May 2007, p. 1902-1909.

[LUA 05] LUA E. K., CROWCROFT J., PIAS M., SHARMA R., LIM S., « A survey and comparison of peer-to-peer overlay network schemes », *Communications Surveys & Tutorials, IEEE*, vol. 7, 2005, p. 72-93.

[MAG 05] MAGONI D., « Network Topology Analysis and Internet Modelling with Nem », *International Journal of Computers and Applications*, vol. 27, n° 4, 2005, p. 252-259.