

SOMMAIRE

| | |
|-----------------------------|------|
| SOMMAIRE | III |
| TABLEAU DES FIGURES : | V |
| DEDICACES | VII |
| REMERCIEMENTS : | VIII |
| RESUME : | IX |

CHAPITRE I :

| | |
|--------------------|----|
| INTRODUCTION | 10 |
|--------------------|----|

CHAPITRE II :

| | |
|---|----|
| LES SYSTEMES DE COORDONNEES DANS L'INTERNET | 13 |
|---|----|

| | |
|---|----|
| II.1. Conception de systèmes de coordonnées réseau : | 13 |
| II.1.1. Les bases théoriques des coordonnées réseau | 13 |
| II.1.2. Structure du système | 13 |
| II.1.3. Les objectifs de performance du système : | 15 |
| II.2. les systèmes de positionnement réseau à base de coordonnées..... | 16 |
| II.2.1. Pourquoi l'utilisation des systèmes de positionnement à base de coordonnées | 16 |
| II.2.2. Systèmes de coordonnées centralisés : | 18 |
| II.2.2.1. La technique Global Network Positioning (GNP) : | 18 |
| II.2.2.2. La technique Lighthouses : | 20 |
| II.2.2.3. La technique Network Positioning System (NPS) : | 22 |
| Vue plus détaillée sur le système NPS : | 24 |
| II.2.3. Systèmes de coordonnées distribuées : | 27 |
| II.2.3.1. La technique Pratical Internet Coordinates (PIC) : | 27 |
| II.2.3.2. La technique Big-Bang Simulation (BBS) : | 28 |
| II.2.3.3. Le système Vivaldi : | 29 |
| II.3. Discussion | 31 |

CHAPITRE III :

| | |
|--|----|
| LIMITATIONS DES SYSTEMES DE COORDONNEES RESEAU | 34 |
|--|----|

| | |
|---|----|
| III.1. Les nœuds balises (Landmarks) : | 34 |
| III.2. Sécurité dans les systèmes de coordonnées: | 35 |
| III.2.1. Sécurité de NPS : | 35 |
| III.2.2. Sécurité de PIC : | 36 |
| III.1. Violation du principe d'inégalité triangulaire: | 38 |

CHAPITRE IV :

| | |
|--|-----------|
| CARACTERISATION DES TIVS DANS L'INTERNET : | 40 |
| IV.1. L'impact de la violation d'inégalité triangulaire sur les systèmes de coordonnées : | 41 |
| IV.2. Les paramètres de détection de TIVs : | 44 |
| IV.2.1. Proposition de métrique pour la détection de TIV : | 46 |
| IV.2.2. Comparaison des paramètres : | 49 |
| CONCLUSION: | 52 |
| REFERENCES: | 53 |



TABLEAU DES FIGURES :

| | |
|--|-----------|
| Figure 1: Structure de systeme de coordonnees reseau | 15 |
| Figure 2: Modele d'espace geometrique d'internet | 17 |
| Figure 3: Methode de positionnement reseau | 18 |
| Figure 4: Methode de positionnement reseau dans espace euclidien a deux dimensions..... | 20 |
| Figure 5: Processus de positionnement avec lighthouses..... | 22 |
| Figure 6: Position inconsistant de serveur..... | 23 |
| Figure 7: Vue d'ensemble de nps | 25 |
| Figure 8: Architecture hierarchique de positionnement reseau | 26 |
| Figure 9: Inegalite triangulaire avec des distances mesurees et estimees dans pic..... | 37 |
| Figure 10: Exemple de violation de l'inegalite triangulaire | 38 |
| Figure 11: Suivi de 3 nœuds d'un triangle abc violant l'inegalite triangulaire tiv | 42 |
| Figure 1 : Relation entre les delais et la severite des tiv (g_a) pour les donnees de p2paim..... | 43 |
| Figure 13: Les courbes tpr/fpr des parametres oreo et rpmo utilisant differents ticks dans p2psim | 47 |
| Figure 14: Les courbes tpr/fpr des parametres oreo et rpmo utilisant differents ticks dans meridian | 48 |
| Figure 15: Les courbes tpr/fpr des parametres ratio de prediction, oreo et rpmo au niveau du reseau king..... | 49 |
| Figure 16: Les courbes tpr/fpr des parametres ratio de prediction, oreo et rpmo au niveau du reseau meridian | 50 |

| | |
|---|-----------|
| Figure 17....: Les courbes tpr/fpr des parametres ratio de prediction, oreo et rpmo au niveau du reseau planetlab..... | 50 |
|---|-----------|

DEDICACES

A ALLAH le Tout Puissant, le Très Haut, le Tout Miséricordieux qui a mis à ma disposition tous les éléments pour la réussite de ce travail et sans qui mon existence n'aurait aucun sens.

A mes chers parents Souleye SANGHARE et Aissatou Gueye SANGHARE, qui m'ont éduqué, aidé et soutenu dans mes études. Je n'oublierai jamais les sacrifices qu'ils ont consenti pour la réussite des mes études.

A ma sœur Absa, mes freres Via Malick, Cheikhna et Pape Makhtar

A mes tantes Alima, Ada, Thiathia, Ndeye Awa, Nene

A mon grand père et ma grand mère Ibrahima Khalil GUEYE et Ndeye Awa SIDIBE

A mes grands parents Ousmane SANGHARE et Khady FALL. Qu'ALLAH les accorde son pardon et les accueille dans son paradis Firdaws

A ma femme Aminata CISS

Que la paix et le salut soient sur notre Prophète Mouhamad.

Qu'Allah nous guide sur le droit chemin, nous pardonne tous nos péchés et que le paradis Firdaws soit notre demeure éternelle. AMINE !

REMERCIEMENTS :

Je tiens à exprimer mes sincères remerciements à toutes les personnes qui, de loin ou de près, ont participé à la réalisation de ce travail.

Mes remerciements vont particulièrement à :

- Mes encadreurs le Dr Bamba GUEYE et le Dr Ibrahima NIANG pour leurs déterminations à vouloir la réussite de ce travail, leurs conseils appréciés, leur aide ainsi que leurs disponibilités durant toute la durée de ce travail.

Je remercie également :

- les membres du jury : le Professeur Cherif BADJI, aux Maitres Assistants le Dr Samba NDIAYE, le Dr Ibrahima NIANG, le Dr Mbaye SENE, le Dr Karim KONATE et le Dr Abdourahmane RAIMY

Je tiens aussi à remercier Youssou Faye, à mes amis du centre de calcul Kasse, Babacar, Ibrahima, Mandicou, Diané, Dame, à Abdallah, à et mes camarades de promotion du DEA 2008-2009 de la section informatique

Je remercie également tous mes frères Salafis, particulièrement ceux des Parcelles Assainies

Résumé :

Au cours de la dernière décennie, de nouveaux services réseaux et d'applications distribués ont émergé. Ces systèmes sont flexibles dans la mesure où ils peuvent choisir leurs chemins de communication parmi tant d'autres. Ce choix de routage s'appuie sur un grand nombre de mesures de délais (RTT) qui sont sources de surcharge dans le réseau. Les systèmes de coordonnées constituent une solution de réduction de surcharge du réseau en optimisant ces mesures directes. Ces systèmes de coordonnées reposent sur une prédiction de distances réseau.

Cependant, les systèmes de coordonnées sont confrontés à des problèmes d'imprécisions dans la prédiction de distance, lorsque les distances mesurées violent le principe de l'inégalité triangulaire (TIV). Plusieurs travaux ont proposé des mécanismes permettant de détecter et de diminuer l'impact des TIVs sur les systèmes de coordonnées. Ces solutions présentent des limites. Soit elles utilisent des quantités de données importantes pour la détection de TIVs, soit elles présentent des inconvénients dans le mécanisme de sélection de voisins dans Vivaldi.

Dans ce mémoire, nous proposons un nouveau paramètre de détection de TIVs appelé "RPMO" qui se base sur le Ratio de Prédiction et la Moyenne des Oscillations des distances estimées. Les résultats obtenus montrent que le paramètre "RPMO" présente des performances meilleures que le paramètre OREE qui représente l'Oscillation et l'Erreur d'Estimation Relative, et le paramètre Ratio de Prédiction.

Mots clés : systèmes de coordonnées, violation d'inégalités triangulaire, prédiction de distance, espace métrique

Chapitre I :

INTRODUCTION

On a assisté ces dernières années, au développement d'une nouvelle classe de services et d'applications à large échelle et globalement distribuées. Nous pouvons en citer comme exemple de services, la publicité ciblée, la sélection automatique de la langue à la connexion, la diffusion de contenu suivant une politique géographique, l'acceptation d'une transaction bancaire seulement à partir d'un endroit pré-établi, les applications de partage de fichiers pair à pair telles que Gnutella [1], OceanStore [2], BitTorrent [3], etc. Ces applications tiennent compte de la topologie du réseau physique, pour la construction de leur propre réseau de recouvrement. En particulier, la plus part de ces systèmes (services, applications) se basent sur la notion de délai pour la localisation d'hôte dans le réseau.

Par exemple, Les applications web décentralisées ont besoin d'information sur la latence, pour associer un client à une localisation, et aussi de guider la sélection de serveur de téléchargement suivant la localisation du client.

Plusieurs approches dans la littérature fournissent des estimations de distances ou de proximité réseau en utilisant des mesures directes entre les paires de nœuds. Parmi ces approches nous pouvons en citer la géolocalisation. Dans cette approche, les techniques de localisation utilisées se basent sur des mesures directes et la connaissance d'adresse IP, pour déterminer la position géographique d'un hôte. C'est ainsi que les techniques présentées dans [4] et dans [5] proposent d'utiliser la position de l'hôte référence, hôte dont on connaît la position géographique le plus proche, en terme de délai comme possible localisation de l'hôte cible. Une autre technique proposée dans [6], permet d'estimer une position en utilisant un nombre suffisant de distances à partir de quelques points immobiles. Cette technique est appelée multilatération. Toutefois pour pouvoir appliquer la multilatération dans l'Internet, il faut que les distances géographiques utilisées soient obtenues à partir des mesures de délai.

Il y a aussi des méthodes basées sur des « traceroute », ou sur l'établissement de positionnement relatif entre les nœuds (l'approche Meridian [7].)

Cependant pour la localisation dans l'Internet, ces approches présentées ci-dessus se basent sur les mesures directes entre paires de nœuds, entraînant ainsi un surcoût de communication,

et conduisant à une consommation élevée de ressource en termes de bande passante. Alors qu'il est important pour les nouvelles applications présentées au dessus, de limiter au maximum possible la consommation de ressources et particulièrement le nombre de mesures à la demande. C'est dans ce contexte, que les Systèmes de Coordonnées dans l'Internet [SCI] (Network Coordinate Systems) ont été proposés, pour réduire la consommation de ressources du réseau, telle que la bande passante. L'idée essentielle d'un SCR est de modéliser l'internet dans un espace géométrique, et de caractériser la position de chaque nœud dans l'internet par une position (c.-à-d. coordonnée) dans cet espace géométrique. Le délai entre toute paire de nœuds est ainsi prédit comme étant la distance géographique entre leurs coordonnées. En conséquence les mesures explicites ne sont plus nécessaires.

Les coordonnées fournissent une approximation des délais internet, et elles ont permis de résoudre un grand nombre de problèmes, tel que la consommation élevée de la bande passante ainsi que la surcharge du réseau causé par les nombreuses mesures directes

En dépit de ces succès, plusieurs problèmes de ces systèmes limitent significativement les avantages des coordonnées, tel que la violation du principe de l'inégalité triangulaire (TIVs) : Prenons l'exemple de trois nœuds A, B et C tels que $d(A,B)$ est de 40 ms, $d(B,C)$ est de 35 ms et $d(A,C)$ est de 100 ms où $d(X, Y)$ représente le délai RTT existant entre le nœud X et le nœud Y. Dans ce cas, le principe de l'inégalité triangulaire est violé car $d(A,C) > d(A,B) + d(B,C)$.

En effet les TIVs causent des problèmes de précision des coordonnées au niveau des délais internet. Des études précédentes [8], [9] ont évalué la précision dans l'utilisation des coordonnées dans la sélection de voisin, et ont montré que les coordonnées ne sont pas assez précises, dans l'article [10], les auteurs montrent que la violation de l'inégalité triangulaire dans les délais internet est une importante cause de l'imprécision des coordonnées.

Cependant beaucoup de questions restent floues sur le phénomène de la violation de l'inégalité triangulaire dans les délais internet. Quelles sont les propriétés des TIVs dans les délais internet ? Comment les TIVs influent sur la performance des coordonnées ? Peut-on réduire l'impact des TIVS dans les systèmes de coordonnées ?

La suite du document sera organisée ainsi, dans le chapitre II, nous passerons en revue les différentes classes de systèmes de coordonnées proposées dans la littérature. Au niveau du chapitre III, nous parlerons de la limitation de ces systèmes de coordonnées et dans le dernier

chapitre, nous étudierons les caractéristiques des TIVs, et proposons un paramètre pour l'amélioration de la détection de TIVs au niveau des réseaux, ce qui permettra aux systèmes de coordonnées d'éviter les TIV dans leur calcul de coordonnées et ainsi améliorer l'exactitude de leur prédiction de distance.

Chapitre II :

LES SYSTEMES DE COORDONNEES DANS L'INTERNET

II.1. Conception de systèmes de coordonnées réseau :

Dans cette section nous donnons un bref aperçu des systèmes de coordonnées et par la suite faisons un état de l'art des différentes techniques qui utilisent les systèmes de coordonnées. Ces techniques peuvent être regroupées en deux catégories : centralisées et décentralisées.

II.1.1. Les bases théoriques des coordonnées réseau

Dans un système de coordonnées, chaque nœud choisie un nombre constant de voisins. Le but d'un tel système est d'assigner à chaque nœud un ensemble de coordonnées, de telle sorte que, les distances entre les coordonnées des nœuds pourront prédire les délais mesurés avec précision.

Les connections entre tous les voisins (nœuds voisins) forment un graphe de voisin $G(V,E)$, où V est l'ensemble des nœuds du réseau, et E l'ensemble des arêtes de voisin. Le calcul des coordonnées peut être généralement formulé comme un problème d'encastrement du graphe. Le problème est de déterminer la localisation x_i de chaque nœud dans un espace à d dimensions, tel que pour chaque pair de voisin (i,j) , la distance entre x_i et x_j peut prédire le délai entre i et j .

II.1.2. Structure du système

Tous les systèmes de coordonnées peuvent être absorbés dans un système de framework général. Comme montré dans la figure 1, un système de coordonnées est composé de quatre composants majeurs : l'architecture, le model, la procédure d'optimisation et les processus assistants.

Architecture : l'architecture est définie comme étant le graphe de voisin utilisé dans un système de coordonnées. Après la procédure de sélection de voisins, tous les nœuds peuvent former un graphe de voisin. Ce graphe de voisin ainsi connu comme l'architecture du système, constitue une base pour la mesure de délai et le calcul de coordonnées. Un système de coordonnées peut utiliser différentes architectures :

- Une architecture centralisée, dans laquelle tous les nœuds peuvent seulement choisir un ensemble fixe de hôtes référence comme ses voisins ;
- Une architecture complètement décentralisé, dans laquelle tous les nœuds peuvent choisir n'importe quels pairs de nœuds comme voisins.

Modèle : c'est le modèle compact que les systèmes de coordonnées utilisent pour se rapprocher de l'espace de délai d'internet, de sorte que la distance entre les coordonnées virtuelles puisse prédire le délai d'internet avec précision. Plusieurs modèles peuvent être utilisés pour assigner des coordonnées virtuelles et estimer les délais d'internet, tels que le modèle des espaces métriques, et le modèle de matrice de factorisation.

Procédures d'optimisation : Les procédures d'optimisation sont des algorithmes utilisés pour calculer les coordonnées virtuelles, ils se basent sur les mesures de délai et sur un modèle particulier, qui peut être le Downhill Simplex Algorithm [11], ou l'algorithme de simulation Big-Bang proposé dans [12].

Processus auxiliaires (assistants) : en plus de ces composants indispensables, un système de coordonnées peut aussi avoir quelques processus assistant, qui vont lui permettre d'améliorer sa performance sur quelques aspects spécifiques. Par exemple une technique de gravité est proposée dans [13] pour réduire les mouvements des coordonnées, et aussi la technique du filtre Kalman, qui a été proposée dans [14] pour la protection du système des attaques malicieuses.

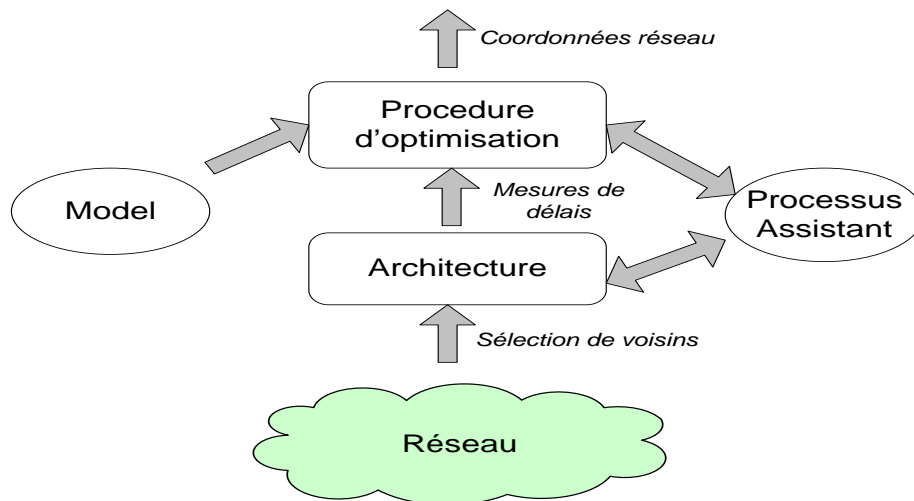


Figure 1 : Structure de système de coordonnées réseau

II.1.3. Les objectifs de performance du système :

Notons que dans la littérature, les chercheurs tentent à atteindre plusieurs objectifs de performance sur la conception de système de coordonnées, tels que :

La Précision : une des applications majeures des systèmes de coordonnées est l'estimation de délai d'internet, ce qui fait que la précision reste un objectif important. En effet un bon système de coordonnées doit pouvoir prédire les délais d'internet, avec précision en utilisant les distances entre les coordonnées des nœuds.

Evolutivité : plusieurs applications, telles que la construction de réseau de recouvrement et le routage géométrique, nécessitent que les systèmes de coordonnées puissent être exécutés dans de très larges réseaux. Dans ce scénario, l'évolutivité devient un objectif non négligeable pour les concepteurs de système. Donc les systèmes de coordonnées doivent être très évolutifs, de sorte qu'on n'assiste pas à des dégradations de performance significatives, s'ils sont exécutés sur de très larges réseaux,

La stabilité : la stabilité des coordonnées est caractérisée par l'oscillation rapide des coordonnées du réseau. C'est-à-dire les mouvements rapides des nœuds du réseau. En effet si les coordonnées présentent des oscillations, cela implique qu'elles périssent facilement et qu'elles ne sont pas gardables.

Quand un programme d'utilisateur veut utiliser les coordonnées d'un autre nœud dans le réseau, il questionnera ce nœud très fréquemment pour avoir la plus récente coordonnée. D'après les perspectives de conception de système, un bon système de coordonnées doit pouvoir fournir des coordonnées stables. Ainsi ces coordonnées seront gardables et pourront être facilement utilisées par les programmes d'utilisateur.

Fiabilité : la fiabilité d'un système fait référence à la performance du système dans des scénarios imprévus. La fiabilité d'un système est spécialement importante lors du déploiement du système dans un environnement réel. Quand le système est exécuté dans un environnement réel d'internet, il est possible que quelques nœuds dans le réseau soient malicieux. Les nœuds malicieux peuvent fournir de mauvais délais, ou de mauvaises coordonnées pour tromper les autres nœuds du réseau. Ainsi les concepteurs de système ont pris en considération la fiabilité des coordonnées. Un bon système de coordonnées doit pouvoir tolérer les attaques malicieuses.

II.2. les systèmes de positionnement réseau à base de coordonnées

Dans cette section, nous présentons une vue d'ensemble des différentes architectures proposées dans le domaine des systèmes de positionnement Internet à base de coordonnées [15]. Elles peuvent être réparties en deux classes principales: Les systèmes de coordonnées basés sur les balises fixes (*landmarks*), et les systèmes de coordonnées distribués. Nous parlerons aussi des violations d'inégalité triangulaire et de leurs effets sur les systèmes de coordonnées.

II.2.1. Pourquoi l'utilisation des systèmes de positionnement à base de coordonnées

Pour combler l'écart entre les objectifs contradictoires d'optimisation de performance et d'évolutivité plusieurs approches de prédiction de distances réseau (délai de propagation et de transmission aller/retour « RTT ») à base de coordonnées ont été proposées. Comme illustré sur la figure 2, l'idée principale de tels systèmes est de modéliser l'internet comme un espace géométrique (exemple : un espace euclidien à 3 dimensions), et de définir la position de chaque nœud dans l'internet par une position dans cet espace.

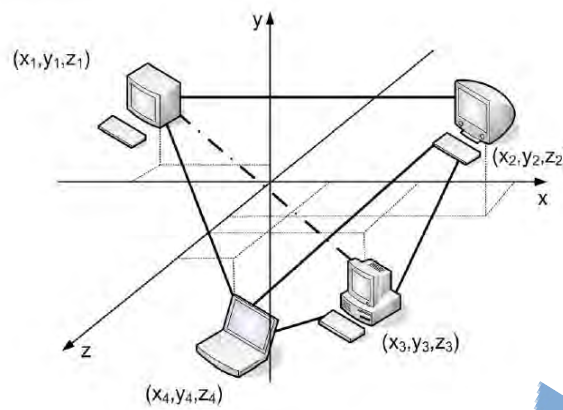


Figure 2 : Modèle d'espace géométrique d'Internet

La distance réseau entre deux nœuds est ainsi prédite comme la distance entre leurs coordonnées sans faire de mesures explicites. En d'autres termes, si un nœud x connaît la coordonnée d'un nœud y , x n'aura pas besoin de faire une mesure explicite pour déterminer le RTT à y ; à la place, la distance entre x et y dans l'espace de coordonnées est une prédiction précise du RTT. En d'autres mots, tant qu'une position précise et raisonnable pour un nœud peut être obtenue avec de petits efforts, beaucoup de coût de mesures de distance par sondage peuvent être éliminées.

En fin de compte, les systèmes à base de coordonnées pour le positionnement réseau visent à pouvoir fournir des positions réseaux à tous les nœuds dans l'internet avec évolutivité et de façon opportun.

Les propriétés de telles approches de prédiction de distances réseau sont les suivantes :

- l'Architecture pair-à-pair : les systèmes à base de coordonnées peuvent être incorporés naturellement dans des applications pair-à-pair, puisque les serveurs terminaux peuvent maintenir facilement les coordonnées qui caractérisent leurs positions dans l'internet.
- Extrêmement rapide : quant un terminal (nœud) découvre l'identité d'un autre terminal dans une application pair-à-pair, leurs coordonnées pré calculées peuvent être ferroutées (*piggybacked*), ainsi les distances réseau peuvent être calculées instantanément par le terminal. Il n'y a pas de délai de communication additionnel et le calcul hors ligne des coordonnées de nœud est aussi très simple et rapide.

- Représentation structurée : les coordonnées géométriques des nœuds décrivent une représentation simple, et très structurée de la topologie complexe d'Internet. Beaucoup d'algorithmes peuvent prendre avantage sur cette structure, pour améliorer les opérations topologiques sur l'internet de manière évolutive.

II.2.2. Systèmes de coordonnées centralisés :

Ces systèmes impliquent un composant centralisé (un ensemble de nœuds balises aussi appelés Landmarks ou Hôtes références), à partir duquel les autres nœuds calculent leurs propres coordonnées, suivant des mesures à cette infrastructure fixe déjà déployée (Figure 3). Le concept du positionnement à base de landmarks est initialement introduit dans [16] pour la conception du système Global Network Positioning (GNP).

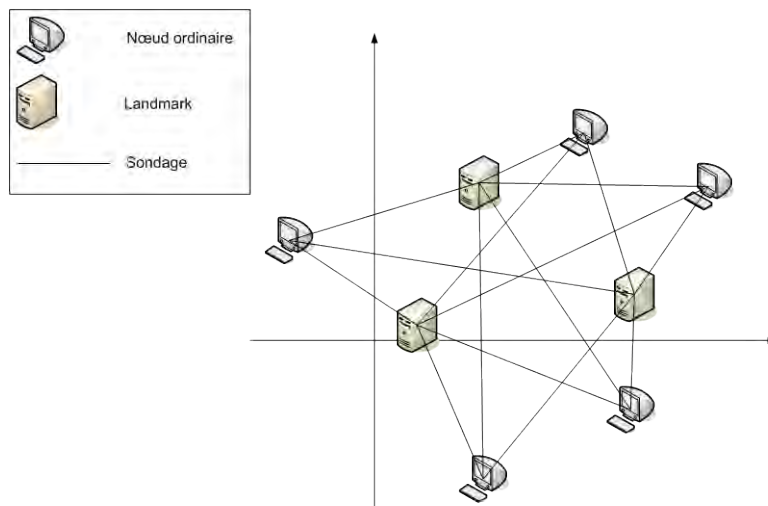


Figure 3 : Méthode de positionnement réseau

II.2.2.1. La technique Global Network Positioning (GNP) :

Dans un tel système [16], les coordonnées des landmarks sont calculées en premier, en minimisant l'erreur entre les distances mesurées et les distances estimées des nœuds balise. Un nœud ordinaire dérive ses coordonnées en minimisant l'erreur entre les distances mesurées et les distances estimées entre lui et les serveurs balises (landmarks). Plus formellement, le

fonctionnement est le suivant: N nœuds spéciaux appelés landmarks, dénotés par LM_1, \dots, LM_N sont déployés dans l'Internet et les distances entre Landmark sont mesurées et transmises à un nœud central. Ainsi le nœud central calcule un ensemble de coordonnées pour chaque landmark et envoie les coordonnées aux landmarks. Les coordonnées des landmark, dénotées par C_{L1}, \dots, C_{LN} sont le résultat de la fonction objective de minimisation suivante :

$$f_{obj1}(c_{LM_1}, \dots, c_{LM_N}) = \sum_{i,j \in 1, \dots, N | i > j} \epsilon(d_{LM_i LM_j}, \hat{d}_{LM_i LM_j})$$

Où $d_{LM_i LM_j}$ est la distance mesurée entre LM_i et LM_j ;

$\hat{d}_{LM_i LM_j}$: est la distance géométrique entre C_{LM_i} et C_{LM_j} , que nous appellerons distance virtuelle ;

$\epsilon(.)$ est la fonction d'erreur de mesure :

$$\epsilon(d_{H1H2}, \hat{d}_{H1H2}) = \left(\frac{d_{H1H2} - \hat{d}_{H1H2}}{\hat{d}_{H1H2}} \right)^2$$

GNP utilise l'algorithme du Simplex Downhill [11] pour résoudre le problème d'optimisation dans les calculs de coordonnées.

Les coordonnées des landmarks définissent les bases de l'espace géométrique. Pour qu'un nœud ordinaire H puisse être encastré dans l'espace géométrique, H utilise les landmarks comme des points de référence et les sonde pour obtenir les coordonnées et les distances réseaux entre lui et chaque landmarks (voir figure 2.2), puis il calcule sa coordonnée C_H que minimise la fonction suivante :

$$f_{obj2}(c_H) = \sum_{i \in 1, \dots, N} \epsilon(d_{HLM_i}, \hat{d}_{HLM_i})$$

L'utilisation de cette méthode avec les mesures d'internet dans [16] montre que, les délais d'internet sont encastrables avec précision dans un espace Euclidien d'une petite dimension. En particulier dans un espace Euclidien à 7 dimensions, 50% des prédictions de distance ont moins de 10% d'erreur, 90% d'entre eux ont moins de 50% d'erreur, et les positions réseaux peuvent être utilisées pour choisir les proches nœuds avec précision dans l'Internet.

Sur ce qui suit, nous présentons différentes extensions du concept des Landmarks introduit par GNP, qui ont été proposé dans la littérature.

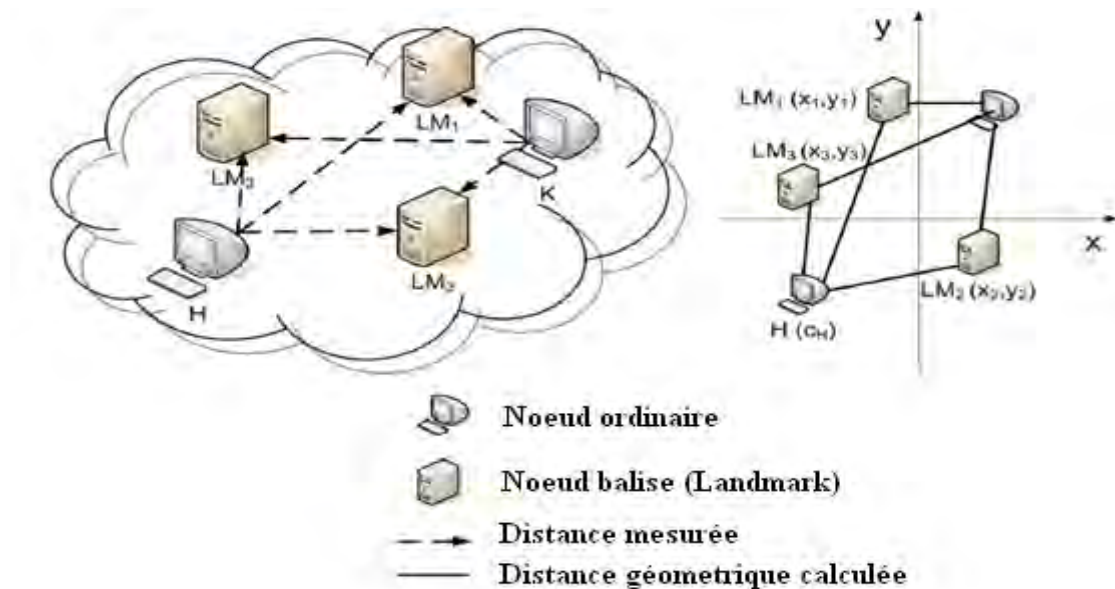


Figure 4 : Méthode de positionnement réseau dans espace Euclidien à deux dimensions

II.2.2.2. La technique Lighthouses :

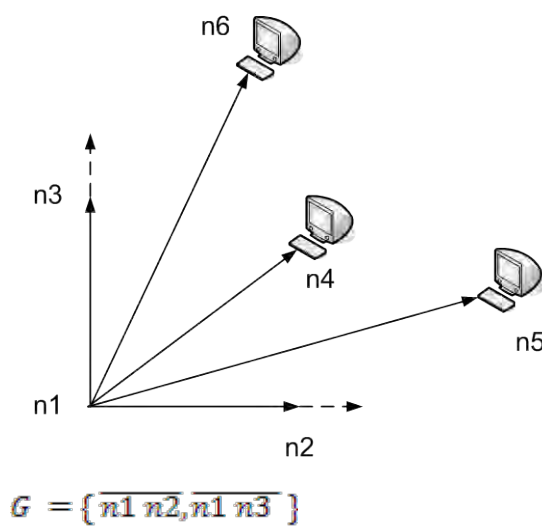
« Lighthouses » est une extension de GNP cherchant à surmonter les limites générées par l'utilisation des landmarks telles que, les congestions, causées par l'augmentation du trafic de mesure vers les landmarks et aussi de pannes de ces derniers. Pour l'aborder, Lighthouses utilise plusieurs ensembles de Landmark, où chaque nœud ordinaire fait les mesures de distances sur un seul ensemble de landmarks. Lighthouse est construit sur le concept de multiples *bases locales* avec une *matrice de transmission*.

Lighthouses [17], comme GNP, a un ensemble de nœuds balise spéciaux appelé landmarks globales. Un nœud A qui joint Lighthouses n'a pas besoin de questionner ces landmarks globaux. En effet, il contacte en premier n'importe quel nœud B, qui est dans le système. Le nœud B fournit au nœud A la liste des nœuds qui peuvent lui servir de landmarks dans un espace géométrique à n dimensions. Le nœud A entrant choisit aléatoirement $n + 1$ nœuds parmi les nœuds qui sont dans la liste. On pourrait noter que si le nœud A ne peut pas trouver les $n + 1$ landmarks, il construit une base locale avec des nœuds ordinaires toujours présents dans le système. Tout nœud ordinaire a sa propre base aussi appelé base

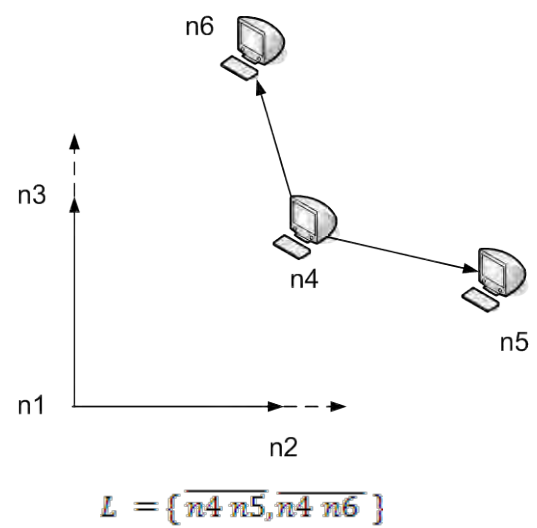
locale, et calcul ses coordonnées en utilisant cette base tandis que dans GNP la même base est utilisée par tous les nœuds ordinaires dans le système. En conséquence, le nœud A construit une base locale $L = \{l_1, l_2, \dots, l_n\}$, où chaque vecteur l est un couple de landmarks, en appliquant la méthode du *Gram-Schmidt* [18]. La méthode du Gram-Schmidt permet de rendre orthogonal un ensemble de vecteurs dans un espace Euclidien.

Pour comparer les coordonnées de deux nœuds, les positions doivent être représentées dans la même base. La matrice de transmission permet de représenter les coordonnées du nœud A dans la base locale d'un autre nœud B. En conséquence, le nœud A calcule une matrice de transmission entre sa base locale et la base globale. Il faut noter que ce processus ne requière aucune mesure de distance additionnelle.

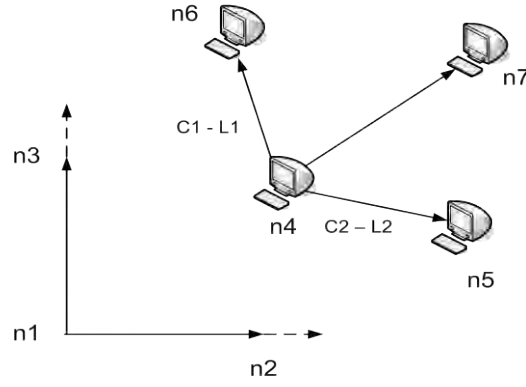
La figure 4 illustre le comportement de Lighthouses dans un environnement à deux dimensions. Considérons six nœuds présents dans le système (n_1 à n_6 figure 4(a)) et un septième nœud n_7 qui veut se joindre au système. La première chose à faire pour n_7 est de contacter un nœud dans le système : par exemple le nœud n_4 . Le nœud n_4 réplique avec une liste de nœuds qui peuvent servir de landmarks pour n_7 , soit $\{n_4, n_5, n_6\}$, n_7 peut ainsi commencer à mesurer la distance entre lui et les Lighthouse. n_7 calcule ensuite sa base locale en utilisant la méthode du Gram-Schmidt [18] (figure 4(b)). Et finalement la figure 4(c) montre le positionnement de n_7 par la matrice de transition.



(a) Recherche de lighthouses



(b) Base locale



$$L1 = \{ \overline{n4 \ n5} \} \quad L2 = \{ \overline{n4 \ n6} \}$$

$$n7 = C1 - L1 + C2 - L2$$

(c) Transition

Figure 5 : Processus de positionnement avec Lighthouses

II.2.2.3. La technique Network Positioning System (NPS) :

Le système NPS [19] étend le système GNP à un système de coordonnées hiérarchique, où certains serveurs peuvent servir de balises « landmarks » (appelés « points références ») pour d'autres serveurs. Le but principal est de pallier aux problèmes de failles des serveurs balises centralisées et fixes, notamment lorsque ces entités deviennent des goulots d'étranglement dans le réseau. La différence majeure par rapport à GNP est que n'importe quel serveur qui s'est bien positionné dans le réseau peut être choisi comme un point référence par un serveur pour d'autres serveurs. Cependant, pour assurer une cohérence dans le système, NPS impose un positionnement hiérarchique entre les serveurs.

Étant donné un ensemble de serveurs, NPS les partitionne dans différents niveaux. Un ensemble de N balises fixes (landmarks) sont placées dans le niveau 0, c'est-à-dire, la couche supérieure du système, et qui définit la base de l'espace géométrique choisi.

Chaque serveur dans un niveau L_i , choisit au hasard quelques serveurs dans le niveau L_{i-1} comme ses points références.

Nous commencerons par la description des objectifs sur la conception de NPS de manière détaillée.

La conception :

L'uniformité – un but important de NPS est de fournir des positions conformes pour les serveurs, c'est-à-dire leurs coordonnées doivent appartenir à la même base. Par exemple, dans la figure 5, c'est insensé de comparer les coordonnées de A et de B puisqu'ils ont des bases différentes. Naturellement si A et B se rendent compte du rapport entre leurs différentes bases, ils peuvent encore corriger la différence.

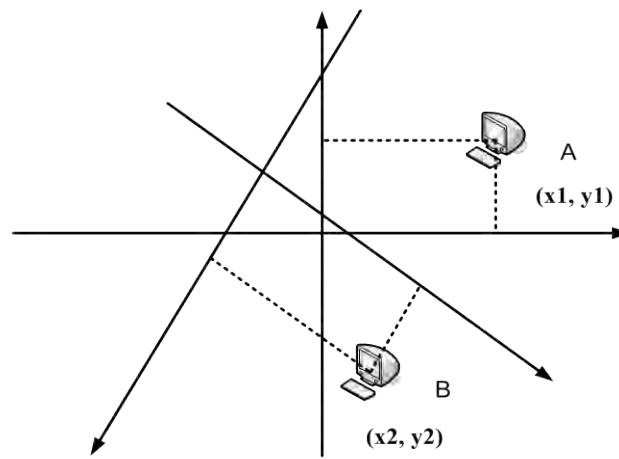


Figure 6 : Position inconsistante de serveur

Adaptabilité – l'internet est un environnement dynamique. A petites échelles de temps, les mesures de distances peuvent être affectées par la fluctuation des temps d'attentes dans les routeurs. A grandes échelles de temps, la topologie d'Internet peut changer. Ainsi les positions des serveurs peuvent avoir besoin d'être mises à jour, pour refléter les changements des distances intrinsèques du réseau. En conséquence un NPS doit s'adapter à la dynamique de l'internet.

La Stabilité – dans un système adaptatif, un problème potentiel est que les positions réseaux peuvent fluctuer même quand il n'y a pas de changement dans la topologie du réseau conduisant à une faible précision. Le but de la stabilité est de minimiser la dérive inutile des positions des centres serveurs, tandis que la réaction de la topologie du réseau change correctement.

Exactitude – l'objectif est de concevoir un système décentralisé qui peut calculer les positions avec l'exactitude qui rapproche étroitement l'exactitude d'une simulation idéalisée de méthode.

Vue plus détaillée sur le système NPS :

La figure 6 illustre les opérations exécutées dans le système. Il y a trois genres de nœuds dans le système NPS : les serveurs d'appartenance (d'adhésion), les serveurs balises, et les serveurs ordinaires.

Les serveurs d'adhésions sont facilement reproductibles. Ils mémorisent les paramètres de configuration du système et maintiennent l'état de quelques serveurs participants. Il peut y avoir plusieurs serveurs d'adhésion indépendants dans le système et il n'est pas nécessaire de synchroniser leur information d'adhésion.

Les nœuds balises sont des nœuds d'infrastructure utilisés pour définir les bases du modèle de l'espace euclidien et peuvent servir de points de référence. Les serveurs balises amorcent leurs positions en utilisant un algorithme décentralisé de positionnement de serveur balise (landmark). Chaque serveur, dans internet peut tourner un processus de NPS qui maintient la position du serveur.

Lorsqu'un processus NPS est démarré, il contacte en premier un serveur d'adhésion (sélectionné par DNS-round robin) pour obtenir un ensemble de paramètres de configuration du système (la dimensionnalité du modèle de l'espace euclidien) et une liste de points de référence (étapes 1 et 2). Alors, le processus NPS commence à sonder les points de référence. Ces sondes ont pour but d'obtenir les dernières positions des points de références et les délais RTT entre le serveur et les points de référence (étape 3). En se Basant sur cette information, le processus NPS calcule la position du serveur (étape 4). Les étapes 3 et 4 sont répétées jusqu'à ce que la position du serveur soit stabilisée. A n'importe quel moment, une application peut questionner le processus NPS pour obtenir la position courante du serveur instantanément (étapes 5 et 6). Excepté que le serveur vient juste d'être amorcé et qu'il n'existe pas d'informations de position gardée antérieurement pour le serveur.

Les serveurs balises et les centres serveurs recalculent périodiquement leurs positions pour s'adapter aux changements de la topologie du réseau.

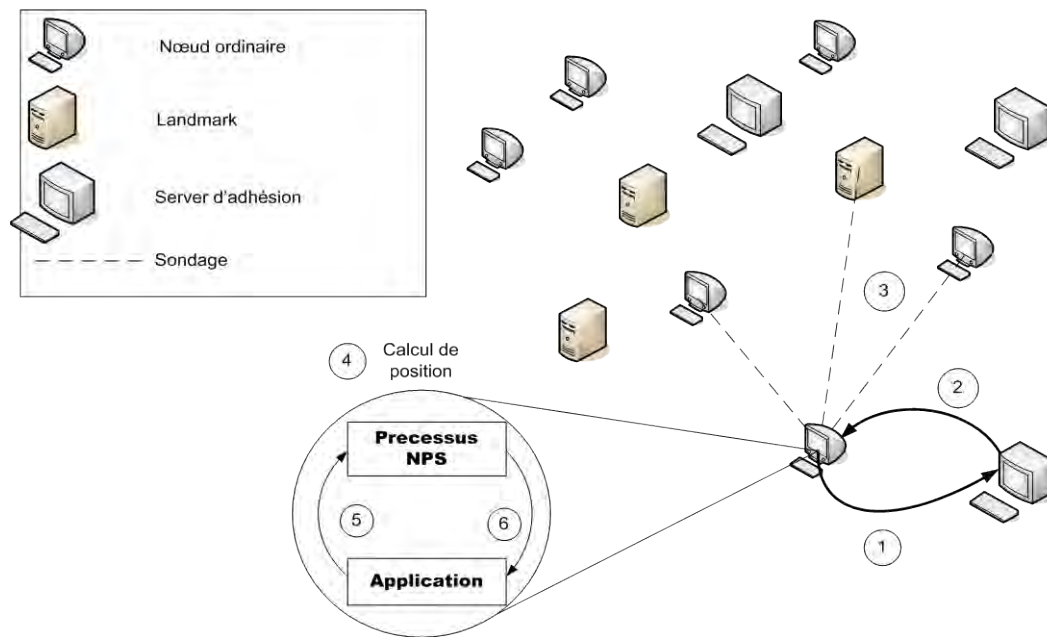


Figure 7 : Vue d'ensemble de NPS

L'architecture hiérarchique :

La conformité de la position est un objectif important. Pour garantir la conformité des positions des serveurs et de leur appartenance aux mêmes bases, une solution est d'utiliser les positions d'un ensemble de serveurs balises pour définir les bases et demander à tous les serveurs d'utiliser les serveurs balises comme leurs points de référence. Bien que cette approche garantisse la conformité, il a quelques propriétés indésirables. Chaque serveur ordinaire sonde tous les serveurs balises pour calculer leurs coordonnées. Dans un tel système, les serveurs balises (landmarks) et leurs liens d'accès de réseau deviennent des goulots d'exécution, et il n'y a pas de solution évidente pour réparer gracieusement les pannes des landmarks.

Ainsi, l'objectif est de trouver une solution décentralisée qui peut en même temps maintenir la conformité. La technique proposée est une architecture hiérarchique. Dans cette architecture, les serveurs balises définissent les bases et peuvent toujours servir de points de référence aux serveurs dans le système. L'exception principale par rapport à l'approche basique est que, chaque serveur qui a déterminé sa position, peut être choisi par le serveur d'adhésion pour être un point de référence pour d'autres serveurs. En conséquence, les serveurs balises deviennent moins critiques, et les pannes temporaires de serveur balise (landmark) n'altéreront pas le système. Présentement, le serveur d'adhésion choisit aléatoirement un serveur parmi les serveurs éligibles comme point de référence, quand les serveurs balises sont

très chargés ou indisponibles. Néanmoins pour garantir la conformité, une dépendance hiérarchique de position est imposée parmi les serveurs. La figure 8 illustre l'architecture hiérarchique.

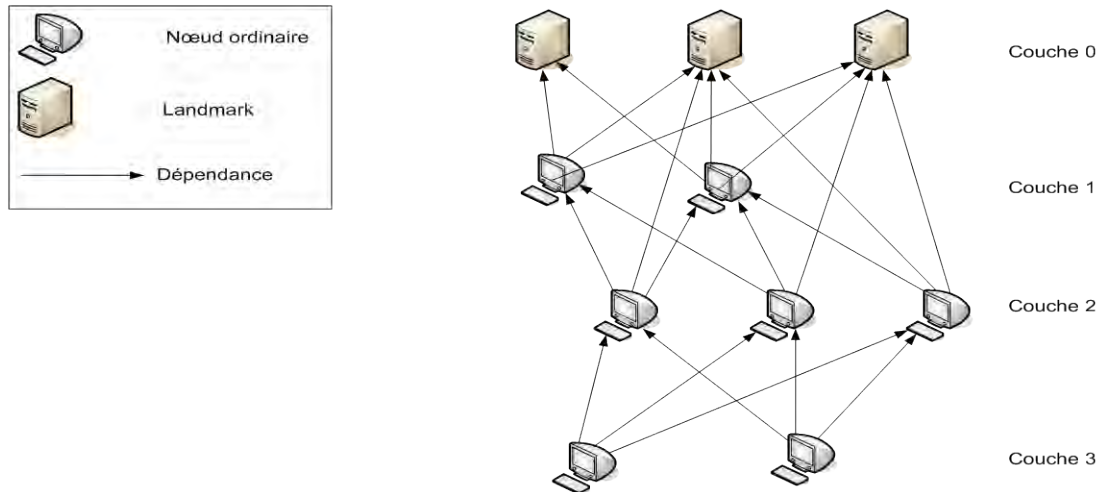


Figure 8 : Architecture hiérarchique de positionnement réseau

On dira que le serveur A dépend du serveur B si A utilise B comme un de ses points de références. Une notion appelée numéro de couche est définie. Le numéro de couche d'un serveur est le nombre maximum de sauts de dépendance le séparant des serveurs balises. Par convenance, la couche numéro 0 est donnée aux serveurs balises.

La dépendance hiérarchique invariante est facile à maintenir dans le système, en contrôlant les numéros de couche des points de référence. Les serveurs balises ont toujours le numéro de couche 0. Quand un processus NPS démarre, il initialise son numéro de couche pour qu'il soit le plus grand numéro de couche alloué dans le système, noté C_{\max} (un paramètre statique récupéré du serveur d'adhésion). Pour maintenir l'invariance, chaque serveur S avec le numéro de couche courant C_S refuse simplement de dépendre de tout point de référence R qui a un numéro de couche $C_R \geq C_S$. Quand S accepte un ensemble de points de référence R_i , S met à jour son numéro de couche pour qu'il soit $\max_i(C_{R_i}) + 1$.

II.2.3. Systèmes de coordonnées distribuées :

Cette approche étend le concept de plongement des nœuds du réseau dans un modèle géométrique, en généralisant le rôle des landmarks à tout nœud présent dans le système, ou en éliminant l'infrastructure de landmark. Les systèmes de coordonnées décentralisés d'internet peuvent être vus comme des systèmes de positionnement réseau pair-à-pair.

II.2.3.1. La technique Practical Internet Coordinates (PIC) :

PIC [20], [21] est un mécanisme basé sur des coordonnées pour estimer des distances dans l'Internet. PIC assigne à chaque nœud un point dans un espace Euclidien à d dimensions et utilise la distance entre deux points dans l'espace comme une estimation de la distance réseau entre les nœuds correspondants. Les nœuds calculent leurs coordonnées dans l'espace Euclidien lorsqu'ils s'intègrent au système. Suivant les coordonnées données pour deux nœuds, n'importe quel nœud peut prédire la distance entre eux.

Contrairement à GNP qui utilise en ensemble de balises fixes (landmarks), dans PIC un nœud rentrant peut choisir n'importe quel nœud du réseau qui possèdent déjà des coordonnées comme nœud balise. Il utilise un protocole de découverte de nœud actif pour trouver un ensemble de nœuds voisin pour le calcul des coordonnées, et s'appuie sur la méthode du Simplex DownHill [11] pour minimiser les erreurs des distances estimées. Exemple, soit N l'ensemble des nœuds dont les coordonnées ont déjà été calculées. Lorsqu'un nœud n se joint au système, il peut choisir n'importe quel ensemble de nœuds L qui est un sous-ensemble de N avec la taille $|L| > d$, où d représente la dimension du réseau dans lequel on travail.

Pour choisir un ensemble de nœuds L , Trois stratégies ont été élaborées:

- **Choix Aléatoire** : les éléments de L sont choisis aléatoirement avec une probabilité uniforme de N ;
- **Choix du plus proche** : les éléments de L choisis sont les éléments de N les plus proches de du nœud n dans la topologie du réseau.
- **Choix Hybride** : certains éléments de L sont choisis aléatoirement et d'autres parmi les plus proche du nœud n

Lors d'un amorçage le comportement du système est un peu différent. Si $|N| < |L|$, n choisit tous les nœuds dans N . Ainsi il obtient les distances mesurées entre tous les paires de nœuds dans N (une matrice $|N| \times |N|$). Dans ce cas, l'algorithme d'optimisation globale calcule de

nouvelles coordonnées pour tous les nœuds dans N en minimisant l'erreur dans les distances prédites entre toutes les paires de nœuds $N \cup \{n\}$.

L'algorithme de Simplex Downhill [11] est utilisé pour calculer les coordonnées comme dans [16]. La fonction d'erreur cible est la somme des carrés des erreurs relatives :

$$e = \sum_{i=1}^{|L|} \left(\frac{d_i^m - d_i^p}{d_i^m} \right)^2$$

Où, d_i^m est la distance mesurée entre le nœud n et le i^{eme} nœud dans L et d_i^p est la distance prédite entre le nœud n et le i^{eme} nœud dans L .

L'intuition derrière les différentes stratégies élaborées pour choisir L dans PIC est la suivante. La stratégie du « **plus proche** » doit fournir à l'algorithme du Simplex Downhill de bonnes informations pour positionner le nœud entrant correctement dans l'espace euclidien par rapport aux nœuds voisins dans le réseau. La stratégie « **aléatoire** » fait la même chose que la celle du « plus proche », par contre elle utilise les nœuds éloignés pour le positionnement correcte du nœud entrant dans l'espace euclidien.

La stratégie du « **plus proche** » doit réaliser de faibles erreurs relatives en prédisant de petites distances, alors que la stratégie aléatoire doit réaliser de faibles erreurs relatives en prédisant de longues distances. Quant à La stratégie « **hybride** », elle doit réaliser quelque chose au milieu c'est-à-dire de faibles erreurs relatives, à moitié avec des nœuds proches et à moitié avec des nœuds éloignés. D'ailleurs Les expériences dans [20] ont montrées que la stratégie hybride réalise de plus faibles erreurs relatives aux deux autres stratégies.

II.2.3.2. La technique Big-Bang Simulation (BBS) :

La technique BBS [12] modèle les nœuds du réseau comme un ensemble de particules. Chaque particule est une image de nœud dans un espace géométrique. Les particules se déplacent dans un espace sous l'effet d'un champ de force potentiel. Le nom de « Big-Bang Simulation » vient du fait que les particules sont initialement placées à l'origine de l'espace. Le champ de force est dérivé de l'erreur d'énergie potentielle qui est égale à l'erreur de plongement totale. Le champ de force réduit l'énergie potentielle des particules et les particules se tirent et se repoussent en se basant sur l'erreur de distance entre elles [22]. Durant chaque phase de calcul, les particules se déplacent suivant des trajectoires tendant à

réduire l'énergie potentielle de tout le système. A la fin de chaque phase, le système atteint approximativement un point d'équilibre où l'énergie potentielle est minimisée.

Dans une phase de calcul à plusieurs itérations où les particules se déplacent dans des intervalles de temps discrets, la position et vitesse de déplacement de particules au temps $t+\delta t$ sont calculées en appliquant les lois de Newton du mouvement, et la nouvelle énergie potentielle est calculée à la fin de l'itération. Notons que les positions des particules et les vitesses calculées dans l'itération courante sont les entrées de la prochaine itération. L'augmentation du *timestep* (étape de temps) δ fournit une plus grande efficacité numérique. Et contrairement, le décroissement du *timestep* permet d'attirer les particules vers une énergie potentielle minimum globale. Une bonne introduction aux lois de Newton du mouvement peut être trouvée dans [22].

Si les particules sont seulement sous l'effet du champ de force, elles vont s'éloigner très rapidement et osciller toujours avec une vitesse constante. Pour réparer ce problème, une force de *friction* est ajoutée (*Friction* est la force de résistance relative au mouvement de deux surfaces en contact, ou d'une surface en contact avec un fluide (par exemple l'air sur un avion)). Avec cette force, une partie de l'énergie est perdue à cause de la friction et après un temps, le système se stabilise. Les particules en mouvement se voient assigner un coefficient de friction μ_k , et les particules statiques sont sous l'effet du coefficient friction μ_s .

La magnitude f_{ij} du champ de force détermine à quelle distance la force induite tire ou repousse les deux particules i et j . Une valeur positive signifie que la force tire les deux particules simultanément (ensemble). Contrairement à une valeur négative où les deux particules sont repoussées. Les auteurs de BBS [12] ont montré que cette force induite est donnée par la dérivée de l'erreur de prédiction.

II.2.3.3. Le système Vivaldi :

Vivaldi [23] est probablement le système de coordonnées le plus réussi. C'est un algorithme simple, distribué pour le calcul de coordonnées. Il ne requière pas de serveurs balises (Landmarks) et fournit une précision similaire à celle de GNP. Chaque nœud dans Vivaldi calcule ses propres coordonnées. Chaque fois qu'un nœud communique avec un autre nœud, il mesure la latence de ce nœud, et ajuste ses coordonnées pour minimiser l'erreur entre les latences mesurées et les latences prédites.

Trouver de bonnes coordonnées conduit à minimiser l'erreur de prédiction entre les latences mesurées et les latences prédites caractérisées par la fonction d'erreur au carré suivante :

$$E = \sum_i \sum_j (L_{ij} - \|x_i - x_j\|)^2$$

Où L_{ij} est le délai aller-retour entre le nœud i et le nœud j , x_i et x_j les coordonnées assignées respectivement à i et à j , $\|x_i - x_j\|$ la distance entre les coordonnées des nœuds i et j dans l'espace de coordonnées choisi.

Vivaldi se base sur un réseau de simulation de ressorts physique pour minimiser E , où la position d'un nœud correspond à celle de l'extrémité d'un ressort. Conceptuellement, Vivaldi place « un ressort » entre chaque paire de nœuds (i, j) dans le système, avec une longueur « au repos » correspondant à la mesure RTT entre ces deux nœuds. La longueur réelle du ressort est considérée comme étant l'estimation de distance entre les deux positions des nœuds. L'énergie potentielle d'un tel ressort est proportionnelle au carrée de déplacement par rapport à sa longueur au repos. Cette énergie représente l'erreur entre le RTT estimé et mesuré. La somme de ces erreurs à travers tous les ressorts est la fonction d'erreur E , que Vivaldi tente de minimiser.

Puisque la fonction d'erreur au carré est équivalente à l'énergie du ressort, elle est réduite au minimum en simulant les mouvements des nœuds sous les forces de ressort.

En effet, Vivaldi utilise une série de petites étapes de temps *timestep* pour les déplacements des nœuds. A chaque étape de temps, la force sur chaque nœud est calculée et le nœud se déplace dans la direction de la force. Le nœud se déplace d'une distance proportionnelle à la force appliquée et la durée du timestep. A chaque fois qu'un nœud se déplace il diminue l'énergie du système, ce qui entraîne la diminution de l'erreur de prédiction.

Dans l'algorithme de Vivaldi décentralisé [23], une procédure identique tourne sur tous les nœuds Vivaldi. Celle-là est basée sur des échantillons récoltés par les nœuds, qui fournissent les informations sur leurs positionnements. Un échantillon utilisé par un nœud i vers j , est ainsi constitué : de la mesure du nœud i vers le nœud j , de la coordonnée du même nœud j , ainsi que de l'erreur locale reportée par j .

L'algorithme traite les nœuds à erreur élevée, en assignant des poids à chaque échantillon collecté. L'erreur relative de cet échantillon, e_e , est alors calculée comme suit :

$$e_e = \frac{\|x_j - x_i\| - RTT_{mesuré}}{RTT_{mesuré}}$$

où x_i et x_j représentent respectivement les coordonnées des nœuds i et j . Le nœud i met à jour son erreur locale comme suit : $e_i = e_e \times w_e + e_i \times (1 - w_e)$, où $w_e = e_i / (e_i + e_j)$ est le poids de l'échantillon calculé à partir de son erreur locale e_i , et de l'erreur locale e_j du nœud j . Le nœud utilise aussi ce poids pour calculer la valeur $\delta_e = C_e \times w_e$, avec $0 < C_e < 1$ qui définit l'amplitude du déplacement du nœud dans la direction spécifiée par cet échantillon. A l'aide de δ_e , Le nœud met alors à jour sa coordonnée locale comme suit :

$$x_i = x_i + \delta_e \cdot (RTT(i,j)_{mesuré} - \|x_i - x_j\|) \cdot u(x_i - x_j)$$

où $u(x_i - x_j)$ est un vecteur unitaire donnant la direction de déplacement du nœud i .

Vivaldi propose également une variante des coordonnées euclidiennes pour améliorer la modélisation des latences d'internet, en présentant la notion de Hauteur [23]. Un espace de Hauteur consiste à augmenter une coordonnée euclidienne avec un vecteur de Hauteur. Ce vecteur modélise la pénalité de latence des liaisons d'accès réseau comme le délai des files d'attente, les lignes de DSL (*Digital Subscriber Line* « Ligne Numérique d'abonné »), ou les modems câblés. Avec la Hauteur, la distance entre les nœuds est mesurée comme leur distance euclidienne, plus la Hauteur qui représente une valeur positive du vecteur d'Hauteur.

II.3. Discussion

Quoique les systèmes à base de coordonnées aient des propriétés avantageuses pour la prédiction de distance sur l'Internet, n'empêche qu'ils aient été critiqués du fait qu'ils disposent d'une précision moins bonne que celle des systèmes à mesure directe. De plus l'existence de TIVs dans l'internet [24] peut être une barrière majeure pour la précision de ces systèmes.

En plus, le choix d'un espace géométrique convenable pour l'immersion de coordonnée, et plus généralement, la modélisation de l'Internet a reçu le plus d'attention au niveau de la communauté de recherche, et a été vu comme un défi majeur. Fondamentalement, les systèmes de coordonnées se concentrent sur de purs espaces Euclidien ou d'autres espaces géométriques simples, comme les surfaces des sphères et **tori**. Dans [25], les auteurs introduisent un nouveau espace de coordonnées qui place les nœuds d'une distance au dessus d'un espace Euclidien (model en hauteur). Shavitt et Tankel [25] ont proposé l'utilisation d'un espace de coordonnées hyperbolique pour modeler l'Internet. Le model hyperbolique

peut adresser un point faible du model en hauteur de Vivaldi. Ce dernier admet que chaque nœud soit derrière son propre lien d'accès. Ainsi, si deux nœuds sont derrière la même grande latence de lien d'accès, le model en hauteur prédiquera de manière incorrecte une grande latence entre les deux nœuds.

Néanmoins, les implémentations et déploiements réels des systèmes de coordonnées d'Internet dans de larges réseaux, montrent que l'utilisation de tels systèmes est avantageuse pour les applications pair-à-pair, et les réseaux de recouvrement ([26], [27], [28]).

Ledlie et al. [13] Ont montré que l'incorporation des coordonnées de Vivaldi dans un million de nœud du réseau Azureus, améliore la capacité de ce dernier. Néanmoins, ceci est atteint en implémentant des techniques spécifiques dans Azureus, afin de supporter les coordonnées d'une façon efficace. Fondamentalement, pour améliorer la précision et la stabilité des systèmes à bases de coordonnées plusieurs techniques ont été proposées :

- Mise à jour de coordonnées à base de gossip, plutôt que les coordonnées soient ferroutées (*piggybacked*) sur des messages du niveau d'application. Cette technique a été montrée pour augmenter la taille de l'espace d'exécution de Vivaldi, en augmentant de l'ensemble des voisins de chaque nœud, et ainsi augmenter sa précision [13].
- L'exclusion de violations d'inégalité triangulaire, c'est-à-dire l'enlèvement d'un petit pourcentage de nœuds avec des violations d'inégalité triangulaire les plus grandes (chapitre suivant) de la matrice de délai d'Azureus. La suppression de 0,5% des nœuds conduit à une amélioration de la précision globale de 20% [13]. Ces observations confirment un travail théorique, qui montre comment diminuer la distorsion d'encastrement en sacrifiant une petite fraction de distances altérées arbitrairement [29]. Ces résultats démontrent principalement que si un mécanisme pourrait empêcher un petit pourcentage des nœuds (violateurs d'inégalité de triangulaire) d'affecter le reste du système, il améliorerait l'exactitude globale.

Alors que le développement des systèmes de coordonnées avec de bonnes précisions reste un challenge à long terme, les approches courantes sont suffisamment précises pour beaucoup d'applications, et fournissent un compromis entre la précision et le coût de mesure pour des réseaux de recouvrement dynamiques.

Mais il faut noter que cela est permis aux dépends de longs temps de convergence, allant de quelques secondes à plusieurs minutes [30]. Ceci est très loin des performances en termes de rapidité de traitement des systèmes de mesures directes entre des paires de nœuds, et en plus

non acceptable pour les applications tenant compte de la topologie, et qui visent à déterminer les « meilleurs nœuds » le plus rapidement possible.

Les systèmes de positionnement à base de coordonnées sont une proposition attractive s'ils sont déployés comme service : ce qui permettra à chaque nœud de faire tourner un système de coordonnées au démarrage du système d'exploitation, et de fournir ainsi des estimations de distance à la demande aux applications et réseaux de recouvrement. Un système de coordonnées est ainsi vu comme une « infrastructure virtuelle », qui supporte une large variété d'applications et de réseaux de recouvrement. En effet beaucoup de systèmes de coordonnées reposent sur des mécanismes de sélection de voisin pour créer leurs structures de réseau de recouvrement. En revanche, ces mécanismes de sélection de voisin sont confrontés aux limitations des systèmes de coordonnées d'internet : en ce qui concerne la protection contre les attaques des nœuds malicieux, qui ont tendance à donner de fausses informations sur leurs coordonnées, et aussi du respect du principe de l'inégalité triangulaire, qui est souvent violée par les délais d'internet. Ces phénomènes créent un environnement étrange qui embrouille les mécanismes de sélection de voisin, et conduit à une imprécision des systèmes de coordonnées sur l'estimation de distance réseau ainsi que le calcul de coordonnées des nœuds.

Chapitre III :

LIMITATIONS DES SYSTEMES DE COORDONNEES RESEAU

Les systèmes de coordonnées d'internet, bien vrai qu'ils sont avantageux en ce qui concerne les prédictions de distances d'internet, ils présentent aussi des limitations que l'on peut voir sur trois aspects : la question des nœuds balises (Landmark), la sensibilité aux violations d'inégalité triangulaire et la sécurité par rapport aux attaques de nœuds malicieux.

III.1. Les nœuds balises (Landmarks) :

L'inconvénient majeur des systèmes à base de landmark est qu'ils nécessitent l'utilisation de landmark pour le déploiement. En fait le nombre de landmarks et leur placement jouent un rôle important en ce qui concerne l'exactitude des prédictions de RTT, et seules les mesures aux landmarks permettent de faire la mise à jour des coordonnées des nœuds. Ce qui fait que les pannes et les surcharges au niveau des landmarks affectent les latences, et conduisent à des mesures de délais très imprécises, et à des coordonnées erronées. Il est bon de noter aussi que les mesures inter-landmark, ralentissent l'évolutivité de tout le système.

Pour le calcul de coordonnées, les systèmes tels que GNP, Lighthouses, et NPS élaborent un problème d'optimisation globale multidimensionnel, qui minimise la différence entre le délai d'internet mesuré et la distance Euclidienne d'un système de coordonnées cartésien. L'algorithme du Simplex DownHill (SDH) est ainsi utilisé pour résoudre le problème de minimisation. Cependant, SDH ne donne que le minimum local proche de la valeur de départ et ne garantit pas que les coordonnées en résultant soient uniques. Ceci mène à l'attribution certaine de différentes coordonnées pour le même nœud selon le processus de minimisation.

Cependant les systèmes distribués, particulièrement évolutifs, surmontent cette limitation en éliminant l'infrastructure de landmark, mais ils sont plus vulnérables aux menaces de sécurité.

III.2. Sécurité dans les systèmes de coordonnées:

Les systèmes de coordonnées sont vulnérables, même à un nombre restreint de nœuds malicieux qui mentent sur leurs coordonnées et leurs mesures. Certains d'entre eux, tels que NPS et PIC incluent une stratégie pour atténuer les effets des attaques malveillantes simples. Ce qui fait qu'ils sont moins vulnérables aux potentiels nœuds malveillants comparés à d'autres techniques de systèmes de coordonnées. En effet, les nœuds malicieux peuvent potentiellement donner de fausses informations sur leurs positions, et/ou gonfler les délais d'Internet en s'appuyant sur des paquets de sonde.

L'idée fondamentale de NPS est d'éliminer un point de référence s'il est mal positionner dans l'espace Euclidien par rapport aux autres points de référence. Malheureusement, le mécanisme de sécurité de NPS peut être défait très simplement. Alors que le mécanisme de détection de PIC, basé sur l'observation que l'inégalité triangulaire est conservée, est affecté par les potentielles violations d'inégalités qui surviennent souvent dans l'internet.

III.2.1. Sécurité de NPS :

NPS inclut une stratégie pour atténuer les effets des attaques malveillantes simples. L'idée fondamentale de NPS est d'éliminer un point de référence, s'il est mal positionner dans l'espace Euclidien par rapport aux autres points de référence. Chaque nœud, en calculant ses coordonnées basées sur des mesures de différents landmarks, rejetterait le point de référence qui fournit une erreur relative sensiblement plus grande que l'erreur médiane de tous autres points de référence. Supposons qu'il y ait N landmarks (L_i), de positions P_{Li} , et les délais d'un nœud H par rapport aux landmarks L_i sont les D_{Li} . Après que H eut calculé une position P_H en se basant sur ces landmarks. Pour chaque L_i , il calcule l'erreur d'ajustement E_{Li} comme $\frac{|distance(P_H, P_{Li}) - D_{Li}|}{D_{Li}}$. Puis le nœud H décide d'éliminer le landmark avec la plus grande E_{Li} .

Les critères employés par NPS pour éliminer un point de référence sont si :

$$\max_i E_{Li} > 0.01 \quad (1) \quad \text{et} \quad \max_i E_{Li} > C \times \text{mediane}_i(E_{Li}) \quad (2)$$

C étant une constante de sensibilité.

Ainsi le landmark qui a le $\max_i E_{Li}$ est filtré (c.-à-d. H essaye de le remplacer par un autre Landmark pour un futur repositionnement).

Malheureusement, il a été montré qu'un tel mécanisme de sécurité est vulnérable aux diverses attaques [30]. Fondamentalement, il consiste du point de vu d'un agresseur, d'interférer les contraintes (1) et (2) en mentant sur ses coordonnées et/ou en altérant les sondes de mesure. Ceci mène à une contradiction entre les latences mesurées et les latences estimées.

III.2.2. Sécurité de PIC :

La PIC vise à défendre la sécurité de son système de coordonnées contre les nœuds malicieux, en employant un test basé sur l'inégalité de triangle.

Fondamentalement, le test se base sur l'observation que l'inégalité triangulaire est maintenue pour la plupart des triplets de nœuds dans l'internet. Par conséquent, soit le triplet de nœuds A, B et C, PIC suppose que, $d(A,B) + d(B, C) \geq d(A, C)$ où $d(i, j)$ est le délai mesuré entre les nœuds i et j , ou la distance virtuelle dans l'espace géométrique.

L'idée derrière le test de sécurité de PIC est la suivante. Un nœud qui ment sur ses coordonnées ou sur sa distance par rapport au nœud joignant le réseau, est susceptible de violer l'inégalité triangulaire. Le nœud joignant le réseau utilise les délais qu'il a mesurés par rapport à chaque landmark et les coordonnées des landmarks pour s'assurer des TIVs. Ainsi il enlève de son propre ensemble de landmarks utilisés pour le positionnement, le landmark qui violent le plus l'inégalité triangulaire.

Soit n un nouveau nœud joignant l'infrastructure de PIC, qui calcule ses coordonnées en utilisant les landmarks i et j . Ceci est illustré par la figure 10. Si $d(n, i)$ est la distance mesurée entre n et i et $d'(i, j)$ est la distance virtuelle estimée (en utilisant les coordonnées) entre i et j , tous les inéquations suivantes devraient se tenir :

$$d(n, i) \leq d(n, j) + d'(i, j)$$

$$d(n, i) \geq d(n, j) - d'(i, j)$$

$$d(n, i) \geq d'(i, j) - d(n, j)$$

la premier inégalité impose une limite supérieure à la distance mesurée à i et les deux autres imposent une limite inférieure de $|d(n, j) - d'(i, j)|$.

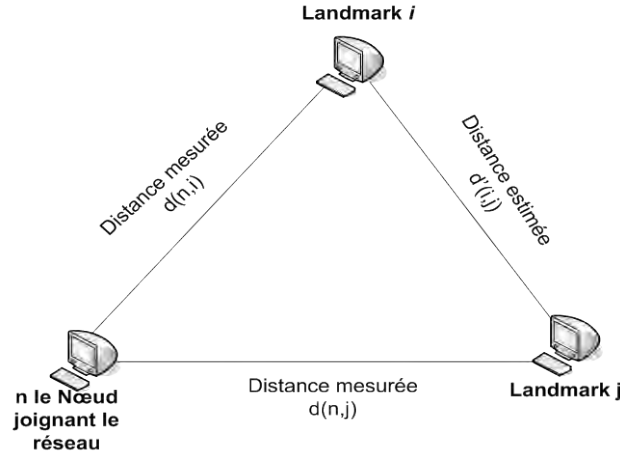


Figure 9 : Inégalité triangulaire avec des distances mesurées et estimées dans PIC

Pour chaque landmark utilisé pour le calcul des coordonnées, le test de sécurité vérifie si les bornes supérieures et les bornes inférieures définies par chaque landmark j sont satisfaites par i , et calcule la métrique $upper_i$ (limite supérieure de i) et $lower_i$ (limite inférieure de i) :

$$upper_i = \sum_{j=1}^N \begin{cases} d(n, i) - (d(n, j) + d'(i, j)) & \text{if } d(n, j) + d'(i, j) < d(n, i), \\ 0 & \text{SINON} \end{cases}$$

$$lower_i = \sum_{j=1}^N \begin{cases} |d(n, j) - d'(i, j)| - d(n, i) & \text{if } |d(n, j) - d'(i, j)| > d(n, i), \\ 0 & \text{SINON} \end{cases}$$

où N est le nombre de landmarks utilisés par n pour le positionnement dans le système. $Upper_i$ est la somme des déviations au dessus des limites supérieures, quant à $lower_i$ c'est la somme des déviations au dessus des limites inférieures. Le test de sécurité calcule la valeur maximale des deux métriques pour tous les landmarks utilisés par n , et enlève le landmark à qui les mesures dévient des bornes supérieures et inférieures calculées. Ainsi le nœud joignant utilise l'algorithme de simplex Downhill pour calculer ses coordonnées avec les landmark restants. Ce processus est répété un nombre de fois fixe.

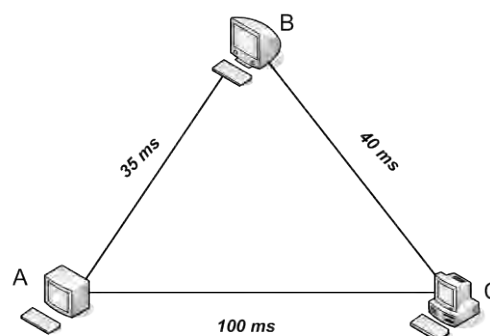
Costa et al. montrent qu'un tel test de sécurité peut traiter jusqu'à 20% des nœuds malveillants existant dans le système [20]. Cependant, les travaux suivants indiquent que RTTs violent généralement et constamment l'inégalité de triangulaire [25] et [24]. Donc un mécanisme de sécurité basé sur le fait que l'inégalité de triangulaire se tient

systématiquement, peut dégrader l'exécution d'un système propre c.-à-d. un système sans nœuds malveillants à l'intérieur.

III.1. Violation du principe d'inégalité triangulaire:

La plupart des techniques de systèmes de coordonnées d'Internet assument que l'inégalité triangulaire est respectée dans l'Internet. En effet les délais ne satisfont pas nécessairement l'inégalité triangulaire due aux politiques de routage, et à l'existence de congestion sur certains chemins, ce qui peuvent entraîner des violations du principe de l'inégalité triangulaire ("Triangle Inequality Violation" - TIV) sur ces délais dans l'Internet [24]. Ce principe stipule que la longueur du chemin direct entre deux nœuds est toujours inférieure à la longueur d'un chemin passant par un nœud intermédiaire. Prenons l'exemple de trois nœuds A, B et C tels que $d(A,B)$ est de 35 ms, $d(B,C)$ est de 40 ms et $d(A,C)$ est de 100 ms où $d(X, Y)$ représente le délai RTT existant entre le nœud X et le nœud Y. Dans ce cas, le principe de l'inégalité triangulaire est violé si $d(A,C) > d(A,B) + d(B,C)$.

Les différentes techniques basées sur des coordonnées vues dans le chapitre précédent souffrent de ces violations d'inégalité triangulaire, du fait que l'immersion exacte des délais dans un espace métrique, où l'inégalité triangulaire doit être respectée est forcément impossible. Face à des situations de TIV, les systèmes de coordonnées les résolvent en forçant les liens à se rétracter ou à s'étirer dans l'espace d'encastrement. Ceci a comme conséquence l'oscillation des coordonnées, et mène ainsi à de grandes erreurs de prédiction de distance.



$$d(A,C) > d(A,B) + d(B,C).$$

Figure 10 : Exemple de violation de l'inégalité triangulaire

Exploiter les systèmes de coordonnées pour diverses opérations de prédiction de distances au niveau applicatif, nécessite que les coordonnées de ces systèmes soient aussi précises que stables.

Chapitre IV :

CARACTERISATION DES TIVS DANS L'INTERNET :

Les recherches sur les violations d'inégalité triangulaire peuvent être regroupées sous deux catégories : les études les performances des systèmes de coordonnées réseau, et les études sur la latence de bout en bout.

Plusieurs études ont examinées l'impact des TIVs sur les performances des systèmes de coordonnées et de positionnement, car ces systèmes tentent de faire l'immersion des délais internet dans un espace métrique où le principe de l'inégalité triangulaire doit être respecté, ce qui n'est pas le cas avec les délais d'internet où on note une présence de TIV causé par les politiques de routage. Ceci Entraîne ainsi des imprécisions sur les calculs de coordonnées de ces systèmes.

Comme les TIVs sont inherents à l'internet nous allons les utiliser à bon escient. Les TIVs peuvent être utiles, en ce qui concerne le routage au niveau des réseaux de recouvrement des applications de distribution de jeux en ligne, de partage de fichier ou de VOIP[31]. Ces applications peuvent potentiellement améliorer leur performance de routage en exploitant les TIVs[32, 33]. En effet, si le coté AB d'un mauvais triangle ABC (un triangle où l'inégalité triangulaire n'est pas respectée) est le lien base de TIV, il existe un raccourci pour aller de A à B en passant par le point C au lieu d'utiliser le chemin direct allant de A à B, les applications ainsi pourront utiliser le raccourci pour gagner plus de temps.

Dans ce qui suit, nous étudions l'impact des TIVs sur le système de coordonnées Vivaldi. Cette étude montre que les nœuds impliqués dans les TIVs sont en général moins précis comparés aux autres nœuds, quant à leur positionnement dans le système de coordonnées. Suivant que ces nœuds sont des parties intégrantes des liens qui violent l'inégalité triangulaire, les calculs des coordonnées de ces nœuds auront tendance à donner de forte amplitude d'oscillations, sans jamais parvenir à des coordonnées précises, ce qui fait qu'ils resteront toujours instables et imprécis. En suite nous regardons la relation qui peut exister entre la sévérité des liens bases de TIVs et leurs distances. En fin nous ferons une comparaison des paramètres de détection de TIV proposés au niveau des articles [34], [35] et

le paramètre que nous avons proposé pour l'amélioration de la détection des TIVs (RPMO). Ces paramètres permettent aux systèmes de coordonnées d'éviter les liens bases de TIV dans leurs calculs de coordonnées, et aussi à certaines applications d'utiliser les TIVs dans le routage au niveau du réseau.

IV.1. L'impact de la violation d'inégalité triangulaire sur les systèmes de coordonnées :

Pour l'étude de l'impact de TIV sur les systèmes de coordonnées, nous utilisons un scénario concret pour montrer comment Vivaldi se comporte quand il y a des violations d'inégalité triangulaire.

En ce concerne nos simulations, nous avons utilisé le simulateur *p2psim* [36] qui fournit une implémentation de Vivaldi. Chaque nœud Vivaldi choisit 32 voisins et la moitié d'entre eux sont choisis parmi les nœuds les plus proches. La constante C_c est fixée à 0,25 (valeur recommandée dans [23]).

Nous avons aussi considéré un espace Euclidien à 9 dimensions, où nous utilisons l'ensemble des données « King » [37] qui contient les RTT mesurés entre 1740 serveurs DNS [36].

Pour les besoins de l'étude de l'impact nous avons pris trois nœuds A, B et C qui forment un triangle A, B, C dans le réseau « King » où la distance $d(A,B) > d(A,C) + d(C,B)$, violant ainsi le principe l'inégalité triangulaire. Puis on exécute Vivaldi sur ces nœuds et récupérons les erreurs d'estimations des distances des liens AB, AC, et CB suivant les 100 derniers pas (TICKS) de la simulation que nous avons représenté sur la figure 11.

L'erreur d'estimation est définie comme étant la différence entre la distance estimée et la distance mesurée (*distance estimée – distance mesurée*)

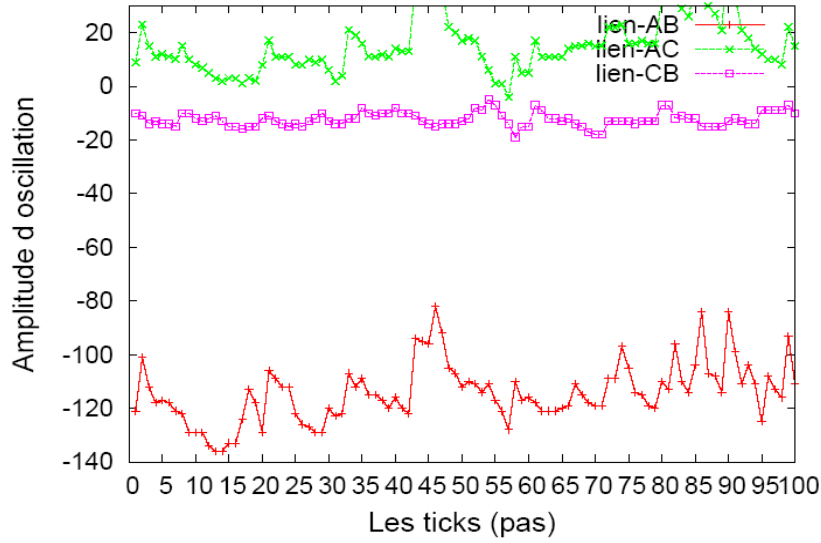


Figure 11 : Suivi de 3 nœuds d'un triangle ABC violant l'inégalité triangulaire TIV

Comme nous pouvons le voir sur la figure précédente. Vivaldi n'arrive pas à trouver des positions exactes pour les nœuds et reste dans des oscillations interminables. Si nous regardons de plus près le comportement du nœud A, qu'est ce qui se passe, en effet à chaque fois que A sonde un voisin B, il ajuste ses coordonnées en se basant sur cette sonde pour décroître l'erreur de prédiction du lien AB. Cependant, à cause de la violation d'inégalité triangulaire dans ABC, il n'existera pas de positions exactes pour les nœuds A, B et C dans l'espace Euclidien pour préserver parfaitement les délais de AB, AC et de CB. Ainsi le résultat est que, tout mouvement de A conduit à décroître l'erreur du lien avec le nœud à qui il a sondé, mais cela n'aide pas à augmenter l'exactitude d'estimation. Tous les nœuds dans le système errent rapidement et l'erreur de prédiction des liens fluctue. Ce résultat montre que, l'existence de TIV peut abîmer l'immersion total du réseau et engendre de très grandes erreurs de prédictions des liens.

Pour l'étude des caractéristiques des TIV, deux critères ont été définis pour indiquer la sévérité des TIVs :

Sévérité absolue : $G_a = d(A,C) - (d(A,B) + d(B,C))$,

Sévérité relative : $G_r = \frac{d(A,C) - (d(A,B) + d(B,C))}{d(A,B)}$.

Ces critères reflètent les gains possibles qu'on peut atteindre en détectant les TIVs. Pour $G_a = 10\text{ms}$ montre que, au lieu de passer par le chemin direct de A à B, passer par le chemin via

le nœud C nous fait gagner 10ms de rapidité. Cependant un grand G_a et G_r ne montrent pas seulement de sévère violation mais aussi un gain possible. Dans notre travail on s'intéresse aux TIVs qui satisfont les deux critères, à savoir $G_a > 10\text{ms}$ et $G_r > 0,1$ et considère les autres comme des non-TIVs.

Pour mieux comprendre les caractéristiques des TIVs, nous avons étudié la relation qui peut exister entre la sévérité des TIV et la longueur des liens. Pour cela, tous les liens dans une matrice de délais sont regroupés dans des bins (groupes) de 10ms en nous basant sur leurs longueurs. Ainsi nous traçons la sévérité des TIVs (G_a) de chaque lien suivant les bins. La figure 12 montre la courbe de la médiane de la sévérité des TIVs contre les longueurs des liens basées sur les données King. Nous remarquons que les liens les plus longs ont tendances à causer de plus sévères violations. Par exemple, les liens qui ont des distances inférieures à 200 ms causent souvent de très légère violation, et les liens dont les distances sont supérieures 300 ms causent de plus en plus de sévères violations, donc proposent des gains de temps plus important au niveau des raccourcis.

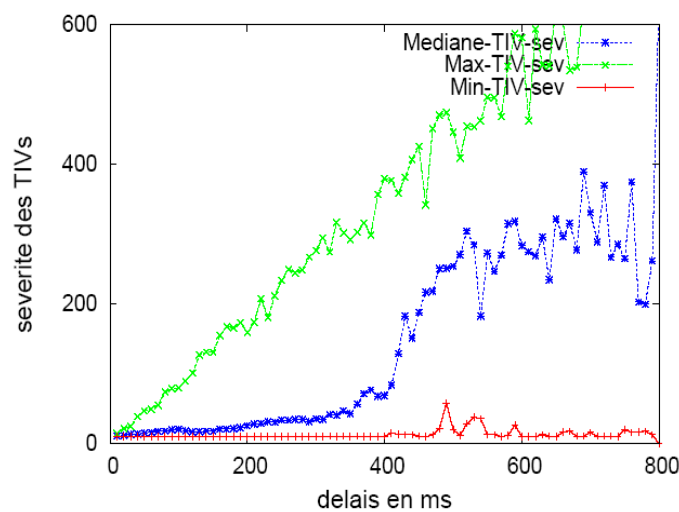


Figure 12 Relation entre les délais et la sévérité des TIV (G_a) pour les données de P2paim

Collection de données :

Pour nos simulations, nous nous concentrons sur l'algorithme de Vivaldi, qui simule un système de ressort et cherche à minimiser son énergie. L'optimisation est très distribuée et est faite sur chaque nœud par itération. Suivant [23], chaque nœud communique avec 32 nœuds voisins, parmi eux 16 sont des nœuds proches et les autres sont choisis aléatoirement. A chaque itération, chaque nœud met à jour ses propres coordonnées en minimisant l'erreur d'encastrement avec ses voisins. Après quelque itération, dépendant de la complexité du

réseau, les coordonnées des nœuds deviennent stables, c'est-à-dire l'erreur d'encastrement de tout le réseau devient aussi petite que le seuil, avec un peu d'agitation.

Pour la collecte de données d'étude, on utilise les jeux de données de King où nous avons 1740 nœuds, Meridian [38] avec 2500 nœuds et PlanetLab [39] avec 180 nœuds, que nous ferons tourner dans le simulateur P2psim qui implémente une version de Vivaldi, puis enregistre les coordonnées des nœuds, après qu'ils soient stables, à K différent temps, pas ou ticks. Sauf contre indication, la valeur de k est fixée à 100 sur toutes nos expériences. Son effet sera discuté dans les sections suivantes. Ainsi à partir des coordonnées obtenues, nous avons calculés les variations de distance dans le temps entre chaque paire de nœuds voisins. Nous nommons la distance mesurée par d et la distance estimée par \hat{d} , et pour chaque lien nous aurons alors $\{d, \hat{d}_1, \hat{d}_2, \dots, \hat{d}_k\}$. Pour k distances estimées nous calculons quelques statistiques tel que :

La moyenne des oscillations $\hat{d}_{moyenne} = moyenne \{ |\hat{d}_1 - d| + |\hat{d}_2 - d| + \dots + |\hat{d}_k - d| \}$,

ou

La variance des distances $\hat{d}_{variance} = variance\{\hat{d}_1, \dots, \hat{d}_k\}$. etc ...

En ce qui concerne nos travaux seuls les liens base de TIV et les liens non Base de TIV nous intéresse, qui peuvent facilement être obtenus à partir des mesures de distance, c'est pour cela que nous nous sommes intéressés sur le pourcentage de lien base de TIV présent dans les ensembles de données King, Meridian et PlanetLab. Nous avons eu à déterminer que 23% des liens dans le réseau King sont des liens base de TIV, 42% liens dans Meridian et 9% des liens dans PlanetLab.

IV.2. Les paramètres de détection de TIVs :

Suivant que les TIVs sont des phénomènes à part entières du réseau d'internet. Notre objectif est alors de trouver des mécanismes de détection des liens bases de TIV présentent dans le réseau. Ce qui permettra aux systèmes de coordonnées de pourvoir les éviter lors des calculs de coordonnée, ou bien de permettre aux applications de pouvoir les utiliser pour le routage dans Internet.

D'après notre étude sur les TIVs, deux paramètres de détection de TIV ont été proposés :

- **Le Ratio de Prédiction** [34]

- **Le paramètre OREE** Oscillation et Erreur d'Estimation Relative (Oscillation and Relative Estimation Error) [35].

Le ratio de prédiction est la relation qui existe entre la distance estimée et la distance mesurée : $\frac{\text{Distance estimée}}{\text{distance mesurée}}$.

Dans [34] les auteurs ont montré que les cotés qui ont de petit ratio de prédiction, c'est-à-dire les cotés qui sont rétrécis dans l'espace euclidien, ont tendance à causer de sévères TIV.

Mais il présente un inconvénient sur la mise à jour des voisins d'un nœud A. En effet un nœud du réseau choisit périodiquement au hasard 32 autres voisins qu'il ajoute sur ses 32 voisins déjà disponibles. Les 64 voisins sont ainsi triés suivant la valeur de leur ratio de prédiction. Le nœud A élimine de sa liste de voisins les 32 nœuds ayant les prédictions de ratio les plus petits. en effet si un lien a un ratio de prédiction très petit, ceci implique que le lien est sous estimé et peut probablement causé des TIVs sévères.

Ce processus répété conduit à ne garder que des voisins proches. Dans ce cas Vivaldi perd de la performance, car il n'a plus de voisin hétérogène c'est-à-dire des voisins proches et éloignés.

Le paramètre OREE représente l'Oscillation et l'erreur relative d'estimation (Oscillation and Relative Estimation Error), ce paramètre met en jeu la variance des distances estimées, la distance mesurée et la moyenne des distances estimées, et les auteurs de [35] ont montré que quand la valeur de OREE est petite le lien sera considéré comme base TIV, et vice versa. Ce qui veut dire que la probabilité qu'un lien soit une base de TIV croît quand la valeur du OREE décroît:

$$\text{OREE} = \frac{\text{variance distances estimées} - \text{distance réelle}}{\text{moyenne distances estimées}}$$

$$\text{Variance} = ((\sum_{i=1}^n di^2) / n) - \overline{d^2}$$

Formule générale :

$$\text{OREE} = \frac{(((\sum_{i=1}^n di^2) / n) - \overline{d^2}) - \text{RTT}}{((\sum_{i=1}^n di) / n)},$$

où d représente la distance estimée d'un lien suivant un pas de simulation (tick).

L'inconvénient du paramètre OREE est qu'il utilise un nombre important d'informations pour la détection de TIVs [35]. L'utilisation de cette technique de détection de TIVs dans de très

larges réseaux tel que l'internet, entraîne un temps de calcul considérable, causant ainsi une contre performance des applications pair-à-pair telles que les applications de jeu en ligne ou de VOIP[31], qui visent à déterminer le meilleur chemin le plus rapidement possible.

IV.2.1. Proposition de métrique pour la détection de TIV :

Suivant ces deux paramètres de détection de TIVs, nous avons eu à proposer un autre paramètre mettant en relation l'oscillation, la distance estimée et la distance réelle d'un lien en vue de mieux détecter les TIVs. Ce dernier est nommé **RPMO** (Ratio de Prédiction sur la Moyenne des Oscillation).

$$\mathbf{RPMO} = (Distance\ estimée / RTT) / Moyenne\ des\ oscillations$$

Pour le calcul de la moyenne des oscillations on prend en considération les variations des distances estimées d'un lien que l'on divise par le nombre de ticks-1.

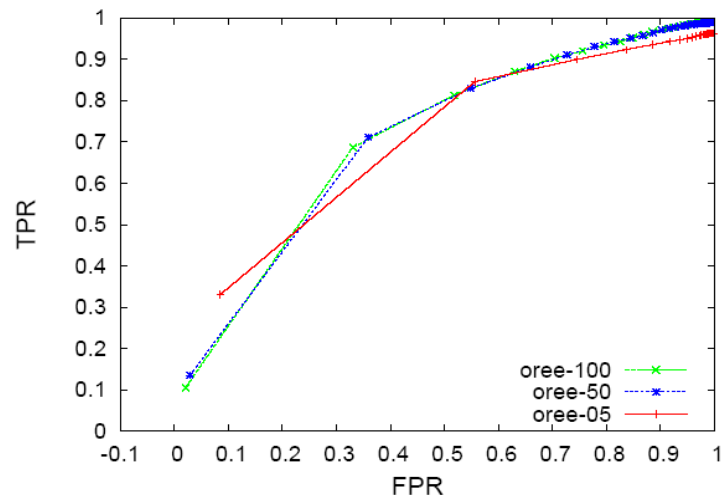
Soit $d1$ la distance estimée de AB au tick1, $d2$ la distance estimée de AB au tick2 et $d3$ la distance estimée de AB au tick3

$$Moyenne\ des\ oscillations\ de\ AB = (|d1-d2| + |d2-d3|)/2.$$

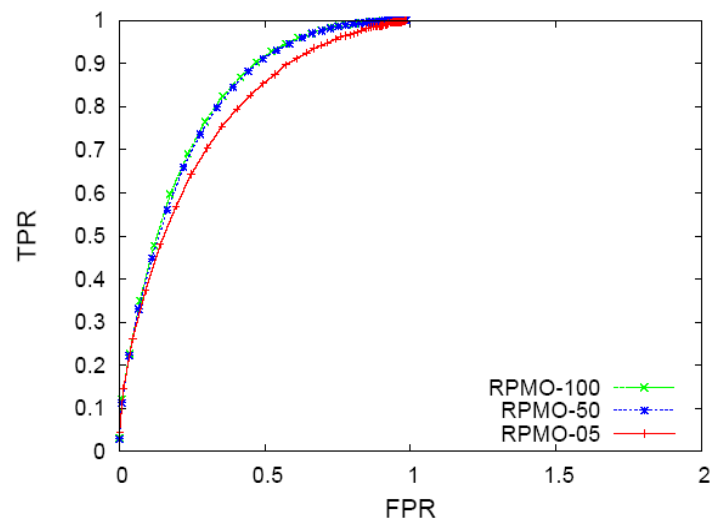
$$\mathbf{RPMO} = (dn / RTT) / ((\sum_{i=1}^n |di - di+1|) / (n - 1)).$$

Le nombre de ticks à utiliser ? Pour l'étude du paramètre K, c'est-à-dire le nombre de ticks à utiliser pour le calcul des paramètres de détection de TIV, qui peut jouer sur la précision des paramètres de détection de TIV à savoir RPMO et OREE. Nous avons fait nos expériences sur les deux ensembles de données, à savoir king où nous avons fait tourner le protocole Vivaldi et Meridian.

Au niveau du réseau King :



(a) OREE

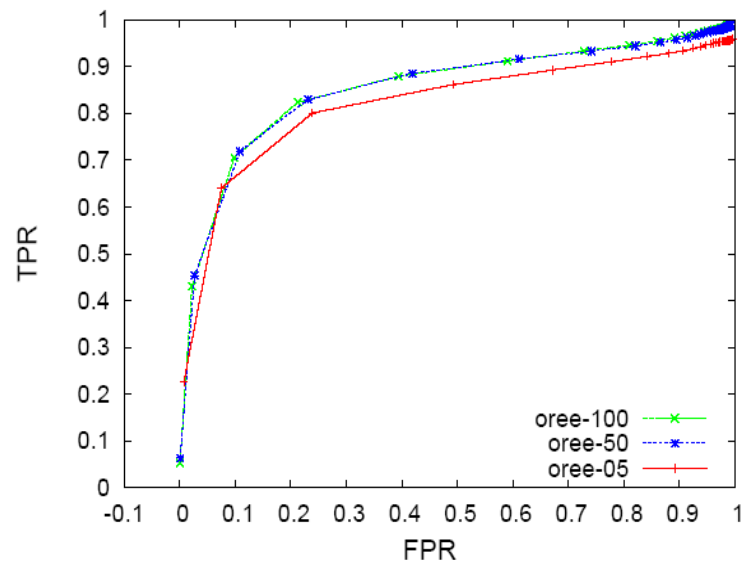


(b) RPMO

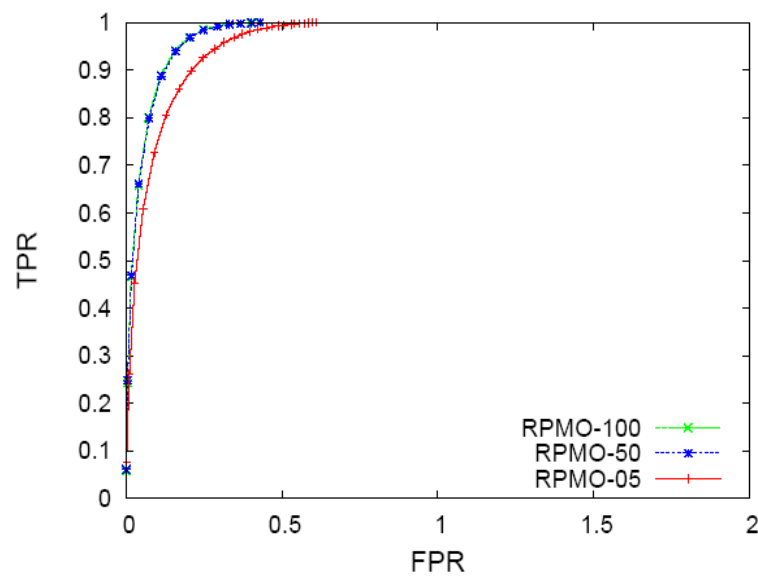
Figure 13 : Les courbes TPR/FPR des parametres OREE et RPMO utilisant differents ticks dans P2psim

Dans P2psim la figure13(a) et la figure13(b) montrent que pour une valeur de $K = 5$, nous obtenons de bonnes performances des deux paramètres de détection de TIV (OREE et RPMO), mais inférieure pour des valeurs de $K = 50$ et $K = 100$.

Le réseau Meridian :



(a) OREE



(b) RPMO

Figure 14 : Les courbes TPR/FPR des parametres OREE et RPMO utilisant differents ticks dans Meridian

Dans le réseau Meridian, la figure13(a) et la figure13(b) montrent les mêmes remarques par rapport aux résultats observés dans le réseau King.

En conclusion, nous pouvons dire qu'avec une grande valeur de K la moyenne des oscillations des distances et la variance des distances deviennent de plus en plus stables, ce qui conduit à l'amélioration de la performance de détection de TIV des paramètres OREE et RPMO. Cependant on note une petite différence pour un K allant de 50 à 100, ce qui nous a poussés à prendre la valeur 100 pour le paramètre K dans nos expériences.

IV.2.2. Comparaison des paramètres :

Pour bien étudier les performances de ces différents paramètres de détection de TIVs (RPMO, Ratio de prédiction et OREE) et pour en déterminer le meilleur, nous avons fait leur comparaison de leurs courbes TPR (True Positive Rate)/ FPR (False Positive Rate) suivant les datasets (King, Meridian et PlanetLab),

Les courbes TRP/FPR nous permet de déterminer le pourcentage de lien base de TIV détecté, et le pourcentage de lien non base de TIV suivant une valeur du paramètre de détection de TIV. Le TPR « pourcentage de vrai positif » concerne les liens base de TIV détectés et le FPR « pourcentage faux positif » concerne les liens non base de TIV détectés comme bases de TIVS.

ce qui nous a donné les résultats suivants.

Pour King :

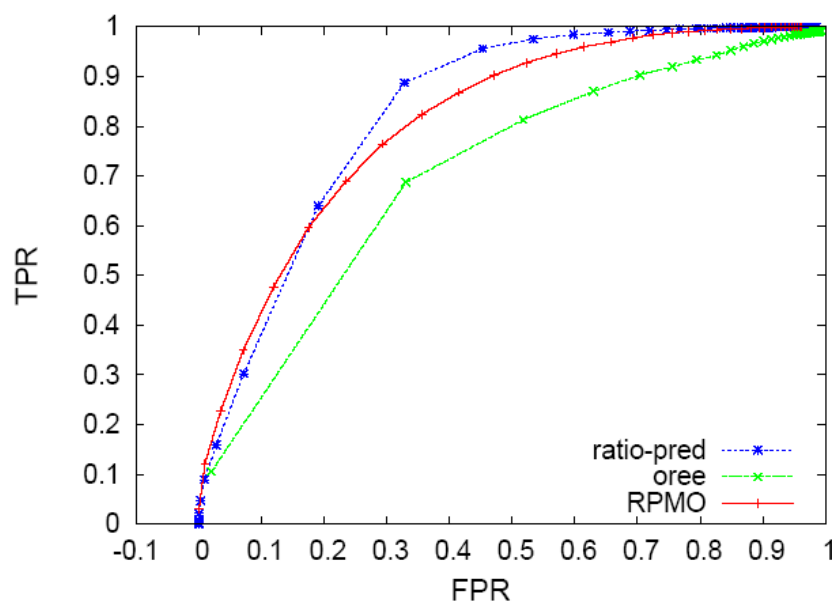


Figure 15 : Les courbes TPR/FPR des paramètres Ratio de Prédiction, OREE et RPMO au niveau du réseau King

Dans Meridian :

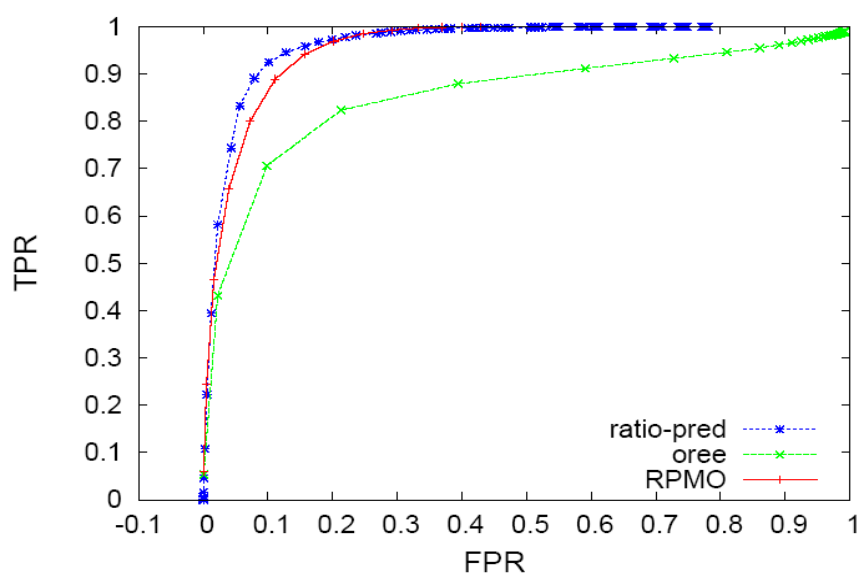


Figure 16 : Les courbes TPR/FPR des paramètres Ratio de Prédiction, OREE et RPMO au niveau du réseau Meridian

Dans PlanetLab :

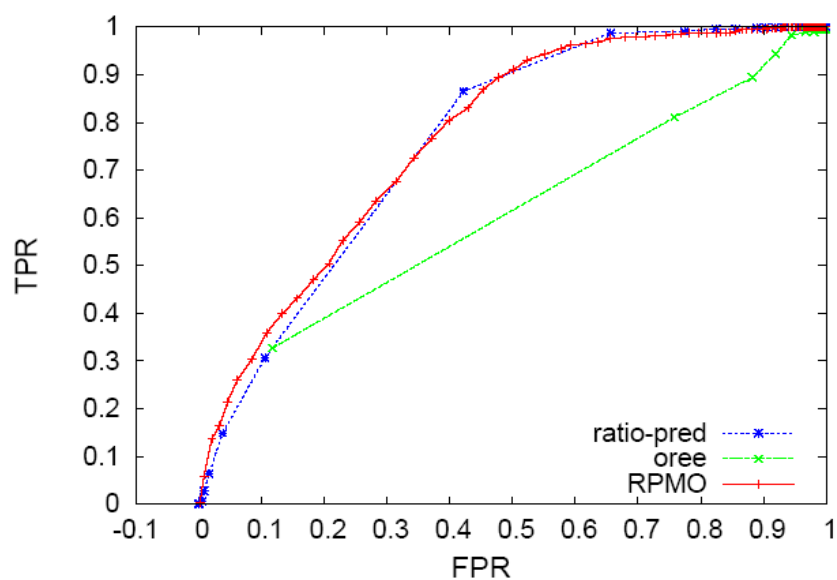


Figure 17 : Les courbes TPR/FPR des paramètres Ratio de Prédiction, OREE et RPMO au niveau du réseau PlanetLab

Suivant ces figures nous pouvons en déduire que parmi les paramètres étudiés, au niveau de King, jusqu'à 60% RPMO est meilleur comparé à OREE et au ratio de prédiction, par contre au delà des 60% le ratio de prédiction devient un tout petit peu meilleur que le RPMO. Dans Meridian nous constatons que le ratio de prédiction et le RPMO sont au coud à coud jusqu'à environ 50%, et c'est là où le ratio de prédiction se détache légèrement du RPMO. Et Dans PlanetLab, les courbes montrent que le RPMO est plus performant que les deux autres métriques. On peut aussi remarquer que le mécanisme de détection de TIV des différents paramètres est plus visible sur le réseau Meridian, ceci s'explique du fait que le réseau Meridian présente un pourcentage plus important de lien base de TIV et de nœuds.

Il apparaît net que notre paramètre de détection de TIV à savoir RPMO est plus performant que le paramètre OREE, et plus performant que le ratio prédiction jusque à un certain pourcentage ou bien même coud à coud en terme de détection de TIV. Mais du fait que le Ratio de Prédiction présente des inconvénients sur le mécanisme de sélection de voisin dans Vivaldi, comme dit précédemment, notre conclusion est que le paramètre RPMO est le plus efficace en matière de détection.

Conclusion:

Les systèmes de coordonnées sont vus comme des approches prometteuses dans la sélection de voisins dans leur efficacité et leur facilité à prédire des délais internet, ainsi que leur distributivité et leur robustesse. L'idée fondamentale de ces approches est de faire l'immersion des délais internet dans un espace géométrique, suivant cette technique d'immersion deux classes de systèmes ont été proposées : les systèmes de coordonnées centralisés et les systèmes de coordonnées distribués qui sont plus tolérants aux pannes et plus évolutives par rapport aux systèmes centralisés qui sont qui par ailleurs un peu plus robuste. Mais ces systèmes sont confrontés à des limites au niveau de la sécurité par rapport aux attaques de nœuds malicieux où seule les systèmes NPS et PIC proposent des stratégies de sécurité contre les nœuds malicieux, ce qui fait qu'ils restent moins vulnérables que les autres systèmes. Nous notons aussi comme limitations des systèmes de coordonnées, les violations du principe de l'inégalité triangulaire, qui entraînent des imprécisions dans la prédiction de délais. Ainsi nous avons proposé un paramètre de détection de TIV appelé RPMO permettant aux systèmes de coordonnées d'éviter ces TIV, ou à certaines applications d'utiliser ces TIV pour le routage. Comparé aux paramètres ratio de prédiction et OREE déjà existant, nous avons conclu, suivant les simulations sur les ensembles de données Meridian, King et Planetlab, que le paramètre RPMO restait le plus performant en matière de détection de TIV.

Cependant il reste difficile de trouver une même valeur de RPMO qui pourra être appliquée dans les ensembles de données considérés dans ce travail. On se charge d'investiguer dans ce sens.

Compte tenu que les TIVs sont nuisibles aux systèmes de coordonnées d'un côté. D'un autre côté les TIVs exposent des opportunités d'améliorer le routage dans le réseau, en offrant des chemins de détournement entre les nœuds avec de petits délais [40]. Ce qui permettra aux applications pair-à-pair, telles que les applications de jeux [41] en ligne ou de VIOP, d'améliorer leur performance en exploitant ces TIVs. L'autre volet de nos perspectives est de faire des recherches sur les techniques de détection des potentiels points C, qui nous permettra d'emprunter les raccourcis des mauvais triangles ABC, tout en sachant que la distance du chemin direct AB est supérieure à la distance des chemins AC et CB, et éventuellement d'implémenter ces techniques.

Références

- [1] Gnutella, "A distributed peer-to-peer data-sharing system," <http://www9.limewire.com/developer/gnutella/protocol/0.4.pdf>.
- [2] J. Kubiawicz et al. "OceanStore: An Architecture for Global-Scale Persistent Storage", in Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), November 2000.
- [3] "Bittorrent specification wiki," October 2008, <http://wiki.theory.org/BitTorrentSpecification>.
- [4] ZIVIANI A., FDIDA S., DE REZENDE J. F., DUARTE O. C. M. B., « Toward a Measurement-based Geographic Location Service », *Proc. of the Passive and Active Measurement Workshop - PAM'2004*, Lecture Notes in Computer Science (LNCS), Antibes Juan-les-Pins, France, avril 2004.
- [5] PADMANABHAN V. N., SUBRAMANIAN L., « An Investigation of Geographic Mapping Techniques for Internet Hosts », *Proc. of the ACM SIGCOMM'2001*, San Diego, CA, USA, août 2001.
- [6] Bamba Gueye, Artur Ziviani, Mark Crovella, Serge Fdida. Vers la Localisation Géographique d'Hôtes dans l'Internet basée sur la Multilatération. (In French) In: *Colloque Francophone sur L'Ingénierie des Protocoles - CFIP'05. pp 415--431*.
- [7] B. Wong, A. Slivkins and E. G. Sirer, *A Lightweight Approach to Network Positioning without Virtual Coordinates*, In Proceedings of ACM SIGCOMM, Philadelphia, August 2005. 14
- [8] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft, "On the accuracy of embeddings for Internet coordinate systems," in *Proc. USENIX Internet Measurement Conference (IMC)*, October 2005.
- [9] R. Zhang, C. Tang, Y. Hu, S. Fahmy, and X. Lin. Impact of the inaccuracy of distance prediction algorithms on internet applications: an analytical and comparative study. In *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2006.
- [10] Sanghwan Lee, Zhi-Li Zhang, Sambit Sahu, and Debanjan Saha. On suitability of Euclidean embedding of Internet hosts. In *Proc. SIGMETRICS 2006*, June 2006.
- [11] J. A. Nelder and R. Mead, *A simplex method for function minimization* Computer Journal, pages 308-313, 1965. 19
- [12] Y. Shavitt and T. Tankel, "Big-bang simulation for embedding network distances in euclidean space," in *Proc. IEEE INFOCOM*, March 2003.
- [13] J. Ledlie, P. Gardner, and M. Seltzer, *Network Coordinates in the Wild*, In Proceedings of NSDI, Cambridge, April 2007. Available as Harvard University Computer Science Technical Report TR-20-06, October 2006. 3, 28, 29, 62, 65, 100, 123, 136, 144
- [14] M.A. Kaafar, L. Mathy, C. Barakat, K. Salamatian, T. Turletti, and W. Dabbous. Securing internet coordinate embedding systems. In *Proceeding of SIGCOMM'07*, August 2007.
- [15] B. Donnet, B. Gueye, M. A. Kaafar : A Survey on Network Coordinates Systems, Design, and Security
- [16] T. E. Ng, and H. Zhang, *Predicting internet network distance with coordinates-based approaches*, In Proceedings of the IEEE INFOCOM, New York, June 2002. 2, 18, 19, 95, 123, 135
- [17] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, "Lighthouses for scalable distributed location," in *Proc. 2nd International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2003.
- [18] L. J. Corwin and R. H. Szczerba, *Calculus in Vector Spaces (Pure and Applied Mathematics): 2nd edition*. Marcel Dekker Inc., dec 1994.

- [19] T. S. E. Ng and H. Zhang, "A network positioning system for the Internet," in *Proc. USENIX Annual Technical Conference*, June 2004.
- [20] M. Costa, M. Castro, R. Rowstron, and P. Key, "PIC: Practical Internet coordinates for distance estimation," in *Proc. 24th International Conference on Distributed Computing Systems*, March 2004.
- [21] M. Szymaniak, D. Presotto, G. Pierre, and M. van Steen, "Partial largescale latency estimation," *Computer Networks*, vol. 52, no. 7, pp. 1343–1364, May 2008.
- [22] H. Young, R. Freedman, T. Sandin, and A. Ford, *Sears and Zemansky's University Physics with Modern Physics*. Addison-Wesley, 2000.
- [23] F. Dabek, R. Cox, K. Kaashoek, and R. Morris, "Vivaldi, a decentralized network coordinated system," in *Proc. ACM SIGCOMM*, August 2004.
- [24] ZHENG H., LUA E. K., PIAS M., GRIFFIN T., « Internet Routing Policies and Round-Trip-Times », *Proc. PAM Workshop*, Boston, MA, USA, avril 2005.
- [25] Y. Shavitt and T. Tankel, "The curvature of the Internet and its usage for overlay construction and distance estimation," in *Proc. IEEE INFOCOM*, March 2004.
- [26] Azureus, September 2008, <http://azureus.sourceforge.net/index.php>.
- [27] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A blueprint for introducing disruptive technology into the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 59–64, Januar 2003.
- [28] J. Ledlie, P. Pietzuch, and M. I. Seltzer, "Stable and accurate network coordinates," in *Proc. International Conference on Distributed Computing Systems*, July 2006.
- [29] Y. Bartal, N. Linial, M. Mendel, and A. Naor, "On metric ramsey-type phenomena," *Annals of Mathematics*, vol. 162, no. 2, pp. 643–709, 2005.
- [30] M. Kaafar, L. Mathy, T. Turetti, and W. Dabbous, "Virtual networks under attack: Disruption Internet coordinate systems," in *Proc. ACM CoNEXT*, December 2006.
- [31] Kho, W., Baset, S.A., Schulzrinne, H.: Skype relay calls: Measurements and experiments. In: IEEE Global Internet Symposium. (2008)
- [32] Lumezanu, C., Baden, R., Levin, D., Spring, N., Bhattacharjee, B.: Symbiotic relationships in Internet routing overlays. In: NSDI. (2009)
- [33] Andersen, D.G., Balakrishnan, H., Kaashoek, M.F., Morris, R.: Resilient overlay networks. In: SOSP. (2001)
- [34] Wang, G., Zhang, B., Ng, T.S.E.: Towards network triangle inequality violation aware distributed systems. In: Proc. the ACM/IMC Conference, San Diego, CA, USA (oct 2007) 175–188
- [35] Yongjun Liao¹, Mohamed Ali Kaafar, Bamba Gueye, Francois Cantin, Pierre Geurts, Guy Leduc : Work in Progress: Detecting Triangle Inequality Violations in Internet Coordinate Systems by Supervised Learning *IEEE/ACM Transactions on Networking*, 2009
- [36] A simulator for peer-to-peer protocols. <http://www.pdos.lcs.mit.edu/p2psim/index.html>.
- [37] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. King: Estimating latency between arbitrary Internet end hosts. In Proc. of SIGCOMM IMW 2002, November 2002.
- [38] Wong, B., Slivkins, A., Sirer, E.: Meridian: A lightweight network location service without virtual coordinates. In: Proc. the ACM SIGCOMM. (aug 2005)

- [39] J. Stribling, “Planetlab all pairs ping,” http://www.pdos.lcs.mit.edu/~strib/pl_app/.
- [40] Savage, S., Anderson, T., Aggarwal, A., Becker, D., Cardwell, N., Collins, A., Hoffman, E., Snell, J., Vahdat, A., Voelker, G., Zahorjan, J.: Detour: A case for informed Internet routing and transport. *IEEE Micro* 19(1) (1999) 50–59
- [41] Bharambe, A., Douceur, J.R., Lorch, J.R., Moscibroda, T., Pang, J., Seshan, S., Zhuang, X.: Donnybrook: Enabling large-scale, high-speed, peer-to-peer games. In: *ACM SIGCOMM*. (2008)