

COMPARAISON ANALYTIQUE DE CAP ET NETPOPPS

Ce chapitre établit une comparaison analytique de nos deux contributions, présentés et analysés individuellement aux chapitres précédents. Il s'agit de CAP (Context-Aware P2p system) et NETPOPPS (NETwork Provider Oriented P2P System)

6.1 Analyse comparative et applicative des contributions

Les deux contributions principales de ce mémoire, CAP et NETPOPPS ont été analysé individuellement, dans chacun des chapitres qui leur ont été respectivement dédiés. Dans ce paragraphe, nous abordons une analyse comparative et applicative des deux systèmes.

Nous commençons par présenter les points communs aux deux propositions. En fait, toutes les deux :

- traitent les identifiants (des pairs et des objets), en tenant compte de caractéristiques du réseau sous-jacent, et assurent un équilibre gagnant-gagnant entre les usagers du P2P et les opérateurs du réseau ;
- s'appliquent à toute DHT mise en œuvre dans un réseau P2P existant ;
- exploitent la fonction injective de hachage de la DHT existante ;
- aboutissent à un système multi-niveaux, en définissant des couches secondaires regroupant les pairs en sous-ensembles suivant une ou plusieurs caractéristiques du réseau sous-jacent (ces sous-ensembles sont dits 'zone' dans CAP et 'entité' dans NETPOPPS), et dans l'overlay correspondant à chaque sous-ensemble, il y a donc une DHT virtuelle (ou VDHT) qui se forme ;

- garantissent la récupération d'un objet du sous-ensemble ('zone' ou 'entité') où il a été trouvé et que le chemin de connexion reste entièrement dans le même sous-ensemble ;
- offrent un cloisonnement des ressources, qui réduit le temps de latence des messages et améliore le temps de récupération des données ;
- favorisent la popularisation des données, en les repartageant dans le sous-ensemble local du pair initiateur de la requête, si la requête a été satisfaite dans un autre ensemble (intermédiaire ou global) ;
- s'adaptent à l'utilisation qui en est faite, de par leur généralité.

Cependant, des différences d'optique sont claires entre les deux systèmes : CAP a une orientation utilisateur, alors que NETPOPPS est bien plus conforme aux intérêts, même économiques, de l'opérateur du réseau.

En effet, CAP est configurable et extensible. Le propriétaire d'une ressource peut définir une politique de partage de cette ressource. Chaque pair n'appartient pas nécessairement à un autre *overlay* que l'*overlay* global, et a fortiori n'appartient pas à toutes les zones du système. Chaque zone est indépendante et tout pair peut créer et initier une nouvelle zone conformément à la métrique de performance qu'il souhaite. La garantie de la découverte d'une ressource en un nombre de sauts minimal est donnée par la DHT. Mais la garantie de la sensibilité contextuelle du routage est donnée par le cloisonnement contextuel des ressources et la sémantique conséquente de l'*objectId* à travers la *Hkey*. CAP peut donc être vu comme un système cœur de réseau au-dessus duquel diverses applications peuvent être développées offrant des services à l'utilisateur via une interface conviviale. Par exemple, l'utilisateur pourra choisir et modifier explicitement la valeur de *Hkey*. Celle-ci peut aussi permettre la création explicite de groupes de communication sécurisés ou avec une qualité de service privilégiée. Ces groupes seront alors autogérés sans qu'il y ait besoin d'une entité de contrôle pour en vérifier l'accès. Aussi grâce à la *Hkey*, CAP peut être aussi adapté pour router des requêtes complexes qui s'exécuteront en parallèle dans plusieurs zones.

Cependant, dans CAP, les différents identifiants d'un même élément (pair ou objet) sont indépendants, et un même élément doit gérer autant d'identifiants que de zones où il réside.

Quant à NETPOPPS, il simplifie considérablement la gestion des identifiants. Chaque élément a ses identifiants en relation mathématique simple entre eux. Et un objet peut même garder un même et seul identifiant dans tout le système, si les différents espaces d'identification du système sont de même taille. Ce qui améliore encore plus les performances (diminue les temps de latence et recherche d'un objet, et donc de réponse).

Mais dans NETPOPPS, le cloisonnement des ressources est conforme aux intérêts de l'opérateur du réseau. Un usager ne peut pas décider de la création d'un *overlay* secondaire, donc non plus d'un groupe de communication, surtout avec des usagers d'autres BEs. La structure du système est fixée par l'opérateur qui gère le réseau de façon quasi-rigide, que seule la renégociation des accords inter-opérateur peut modifier (ou la politique intra-opérateur dans le cas où un opérateur gère plusieurs BEs).

De plus, à travers le nœud de contrôle, NETPOPPS offre à l'opérateur du réseau la possibilité d'exercer un contrôle d'accès sur les pairs utilisant ses réseaux ou de leur offrir ses services propriétaires (e.g. des services privilégiés et/ou payant). La porte est ainsi large ouverte à des services commerciaux, des mesures statistiques, etc.

Ainsi, NETPOPPS permet activement de profiter des importantes économies que peuvent générer les architectures P2P dans la gestion des réseaux. Alors que dans CAP, ceci reste une conséquence.

Enfin, les deux systèmes permettent de développer toute une panoplie de nouveaux services, mais dans NETPOPPS, ceci reste le produit de l'opérateur, selon sa volonté et sous son contrôle.

NETPOPPS, comme la solution de service intermédiaire (*oracle service*) [AFS07] et comme P4P [XKS⁺07], met en avant le rôle qu'un opérateur de réseau peut prendre dans l'adéquation d'un réseau P2P à la topologie sous-jacente, l'optimisation du trafic P2P et des différents coûts, et la performance des applications. Cependant, les opérateurs de réseaux,

notamment les ISPs considèrent généralement la topologie des réseaux qu'ils contrôlent comme étant des informations confidentielles [MG08]. Par conséquent, pour réussir à être adopté à très large échelle, une solution de sensibilisation du trafic P2P à la topologie sous-jacente doit fournir une méthode qui aide les applications dans la sélection des pairs sans qu'il y ait besoin de divulguer explicitement la topologie du réseau sous-jacent. Dans cette perspective, CAP est prometteur, en tant que système générique, flexible et orienté utilisateur.

Toutefois, comme dans tout domaine de recherche applicative, une large adoption à très grande échelle d'un système P2P sensible à la structure sous-jacente ne verra probablement pas le jour sans un accord entre les principaux acteurs sur une solution commune basée sur des standards libres de droit [MG08].

6.2 Exemple de l'incidence positive des contributions sur l'efficacité du routage

Nos deux contributions (CAP et NETPOPPS) proposent des mécanismes de regroupement des pairs, et donc de formation d'entités, suivant un modèle économique donné, pour adapter le flux P2P au modèle économique considéré.

Nous avons aussi dans nos deux contributions qu'un pair physique A est identifié par un *nodeId* n_1 au niveau global, et par un *nodeId* n_2 à un niveau local. Les deux nœuds d'identifiants respectifs n_1 et n_2 sont des représentations logiques du nœud physique A. Il en est de même pour tout autre pair physique. Cela signifie que le nœud n_1 stocke deux DHTs :

- une DHT au niveau global : celle d'un pair physique A ;
- et une DHT au niveau local : celle d'un pair physique B.

Il en découle que quand un pair A consulte sa VDHT (et donc change pour cela de *nodeId*), c'est comme s'il voyait la DHT à ce nouveau niveau d'un autre pair B se trouvant dans la même zone/entité locale que lui.

C'est le profil du système qui nous dira que (par exemple) le pair physique A va se retrouver au plan local dans la même zone/entité que le pair physique B. Mais c'est la construction des identifiants qui nous donnera les différents *nodeIds* d'un pair physique A : nous saurons alors si le pair A – qui, au plan global, aura par exemple le *nodeId* 0 – aura au plan local le *nodeId* 0 ou 2 ou autre.

Par souci de clarté, nous nous positionnons face à un système à un seul niveau secondaire au-dessus du niveau P2P global (cf. fig. 6.1) :

- avec CAP, ceci revient à nous positionner dans le cas où le profil du système définit une seule *Hkey* (qu'elle soit donc simple ou composée, mais pas de *Hkey* dérivée) (cf. p. 88) ;
- avec NETPOPPS, ceci revient à nous positionner dans un cas où il n'y a pas de IEs (cf. p. 99).

Et par souci de simplicité, nous nous limiterons au niveau local à deux entités (ou *zone*, selon qu'il s'agisse de NETPOPPS ou CAP).

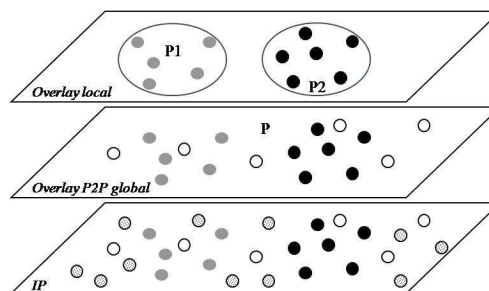


Figure 6.1 : Vue d'ensemble du système analysé

Comme répété plus haut, c'est la méthode de construction des identifiants secondaires qui nous donne les différents *nodeIds* d'un pair physique quelconque. Cette méthode diffère entre nos deux contributions :

- dans NETPOPPS, elle utilise la méthode de dérivation de clés. Et l'on y a distingué deux cas selon qu'un même objet possède un seul ou plusieurs identifiants dans le système, c.-à-d. selon que la fonction injective utilisée pour les dérivations de l'*objectId* primaire est idempotente ou pas (cf. paragraphe 5.2.3 page 104) ;

- dans CAP, la méthode de construction des identifiants secondaires utilise la fonction HMAC (avec une clé $Hkey$ et la fonction de hachage du protocole mis en œuvre), et les identifiants d'un même élément (pair ou objet) sont tous différents et indépendants les uns des autres.

Dans la suite, et par souci de simplicité et de clarté, nous considérons le système analysé comme mettant en œuvre NETPOPPS.

Par ailleurs, nous nous limiterons dans la suite à un espace d'identification de huit éléments, avec sept pairs physiques actifs dans le système, et dont les $nodeIds$ forment l'ensemble $\{0,1,2,3,4,5,7\}$. Tous ces nœuds sont donc présents dans l'*overlay* global. Nous considérons que parmi ces nœuds :

- ceux dont l'identifiant fait partie de l'ensemble $\{0,2,3\}$ sont groupés dans l'entité P1 ;
- et ceux dont l'identifiant fait partie de l'ensemble $\{1,4,7\}$ sont groupés dans l'entité P2.

Aussi considérons-nous que notre système met en œuvre le protocole Chord ([SML⁺01], cf. paragraphe 1.4.2 p. 44).

Dans le cas où les pairs et objets gardent le même identifiant à tous les niveaux (c.-à-d. que la dérivation des identifiants primaires s'applique avec une fonction injective idempotente), la représentation logique de notre système est donc celle représentée par la figure 6.2.

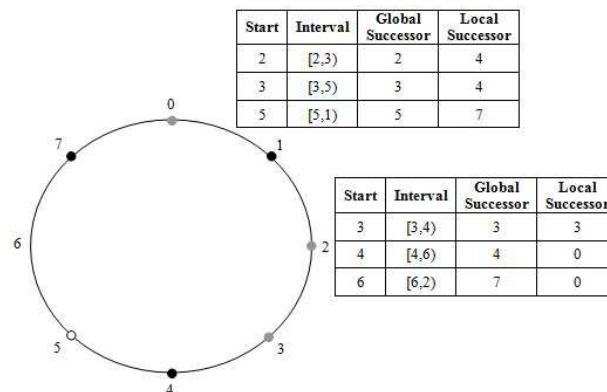


Figure 6.2 : Représentation logique du système étudié

En considérant que chaque élément (pair ou objet) garde le même identifiant à tous les niveaux d'*overlays*, les DHTs logiques du système sont

alors comme données par le tableau 6.1 où N_i représente le nœud logique d'identifiant i .

Tableau 6.1 : Vue globale des DHTs du système (avec une fonction idempotente)

N_0	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>1</td><td>[1,2)</td><td>1</td><td>2</td></tr><tr><td>2</td><td>[2,4)</td><td>2</td><td>2</td></tr><tr><td>4</td><td>[4,0)</td><td>4</td><td>0</td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	1	[1,2)	1	2	2	[2,4)	2	2	4	[4,0)	4	0
Start	Interval	Global Successor	Local Successor														
1	[1,2)	1	2														
2	[2,4)	2	2														
4	[4,0)	4	0														
N_1	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>2</td><td>[2,3)</td><td>2</td><td>4</td></tr><tr><td>3</td><td>[3,5)</td><td>3</td><td>4</td></tr><tr><td>5</td><td>[5,1)</td><td>5</td><td>7</td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	2	[2,3)	2	4	3	[3,5)	3	4	5	[5,1)	5	7
Start	Interval	Global Successor	Local Successor														
2	[2,3)	2	4														
3	[3,5)	3	4														
5	[5,1)	5	7														
N_2	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>3</td><td>[3,4)</td><td>3</td><td>3</td></tr><tr><td>4</td><td>[4,6)</td><td>4</td><td>0</td></tr><tr><td>6</td><td>[6,2)</td><td>7</td><td>0</td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	3	[3,4)	3	3	4	[4,6)	4	0	6	[6,2)	7	0
Start	Interval	Global Successor	Local Successor														
3	[3,4)	3	3														
4	[4,6)	4	0														
6	[6,2)	7	0														
N_3	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>4</td><td>[4,5)</td><td>4</td><td>0</td></tr><tr><td>5</td><td>[5,7)</td><td>5</td><td>0</td></tr><tr><td>7</td><td>[7,3)</td><td>7</td><td>0</td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	4	[4,5)	4	0	5	[5,7)	5	0	7	[7,3)	7	0
Start	Interval	Global Successor	Local Successor														
4	[4,5)	4	0														
5	[5,7)	5	0														
7	[7,3)	7	0														
N_4	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>5</td><td>[5,6)</td><td>5</td><td>7</td></tr><tr><td>6</td><td>[6,0)</td><td>7</td><td>7</td></tr><tr><td>0</td><td>[0,4)</td><td>0</td><td>1</td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	5	[5,6)	5	7	6	[6,0)	7	7	0	[0,4)	0	1
Start	Interval	Global Successor	Local Successor														
5	[5,6)	5	7														
6	[6,0)	7	7														
0	[0,4)	0	1														
N_5	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>6</td><td>[6,7)</td><td>7</td><td style="background-color: #cccccc;"></td></tr><tr><td>7</td><td>[7,1)</td><td>7</td><td style="background-color: #cccccc;"></td></tr><tr><td>1</td><td>[1,5)</td><td>1</td><td style="background-color: #cccccc;"></td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	6	[6,7)	7		7	[7,1)	7		1	[1,5)	1	
Start	Interval	Global Successor	Local Successor														
6	[6,7)	7															
7	[7,1)	7															
1	[1,5)	1															
N_6	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>7</td><td>[7,0)</td><td style="background-color: #cccccc;"></td><td style="background-color: #cccccc;"></td></tr><tr><td>0</td><td>[0,2)</td><td style="background-color: #cccccc;"></td><td style="background-color: #cccccc;"></td></tr><tr><td>2</td><td>[2,6)</td><td style="background-color: #cccccc;"></td><td style="background-color: #cccccc;"></td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	7	[7,0)			0	[0,2)			2	[2,6)		
Start	Interval	Global Successor	Local Successor														
7	[7,0)																
0	[0,2)																
2	[2,6)																
N_7	<table border="1"><thead><tr><th>Start</th><th>Interval</th><th>Global Successor</th><th>Local Successor</th></tr></thead><tbody><tr><td>0</td><td>[0,1)</td><td>0</td><td>1</td></tr><tr><td>1</td><td>[1,3)</td><td>1</td><td>4</td></tr><tr><td>3</td><td>[3,7)</td><td>3</td><td>4</td></tr></tbody></table>	Start	Interval	Global Successor	Local Successor	0	[0,1)	0	1	1	[1,3)	1	4	3	[3,7)	3	4
Start	Interval	Global Successor	Local Successor														
0	[0,1)	0	1														
1	[1,3)	1	4														
3	[3,7)	3	4														

Visualisons à présent chacun des trois *overlays* du système pris individuellement. Commençons par le plan global pris individuellement. Sa représentation logique est alors donnée par la figure 6.3.

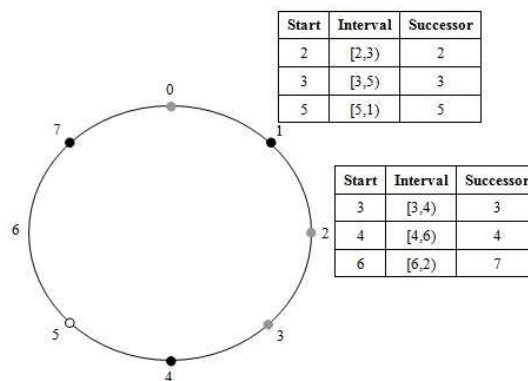


Figure 6.3 : Représentation logique du plan global pris individuellement

Le tableau 6.2 donne les DHTs présentes au plan global.

Tableau 6.2 : Vue des DHTs du plan global

N0	Start	Interval	Successor	Local Successor
	1	[1,2)	1	2
	2	[2,4)	2	2
	4	[4,0)	4	0

N1	Start	Interval	Successor	Local Successor
	2	[2,3)	2	4
	3	[3,5)	3	4
	5	[5,1)	5	7

N2	Start	Interval	Successor	Local Successor
	3	[3,4)	3	3
	4	[4,6)	4	0
	6	[6,2)	7	0

N3	Start	Interval	Successor	Local Successor
	4	[4,5)	4	0
	5	[5,7)	5	0
	7	[7,3)	7	0

N4	Start	Interval	Successor	Local Successor
	5	[5,6)	5	7
	6	[6,0)	7	7
	0	[0,4)	0	1

N5	Start	Interval	Successor	Local Successor
	6	[6,7)	7	
	7	[7,1)	7	
	1	[1,5)	1	

N6	Start	Interval	Global Successor	Local Successor
	7	[7,0)		
	0	[0,2)		
	2	[2,6)		

N7	Start	Interval	Successor	Local Successor
	0	[0,1)	0	1
	1	[1,3)	1	4
	3	[3,7)	3	4

Regardons maintenant l'overlay relatif à P1, pris individuellement. Il est représenté par la figure 6.4

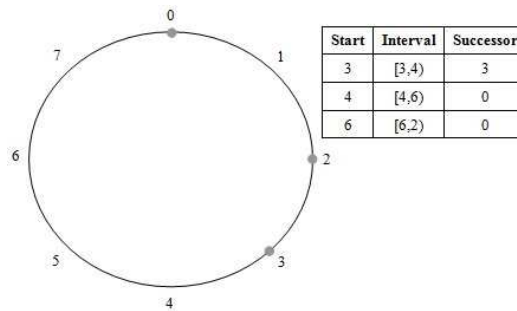


Figure 6.4 : Représentation logique de l'overlay local identifié par P1

Les DHTs relatives à l'overlay correspondant à P1 sont représentées par le tableau 6.3

Tableau 6.3 : Vue des DHTs relatives à l’overlay local identifié par P1

N0	Start	Interval	Global Successor	Successor
	1	[1,2)	1	2
	2	[2,4)	2	2
	4	[4,0)	4	0

N1	Start	Interval	Global Successor	Local Successor
	2	[2,3)	2	4
	3	[3,5)	3	4
	5	[5,1)	5	7

N2	Start	Interval	Global Successor	Successor
	3	[3,4)	3	3
	4	[4,6)	4	0
	6	[6,2)	7	0

N3	Start	Interval	Global Successor	Successor
	4	[4,5)	4	0
	5	[5,7)	5	0
	7	[7,3)	7	0

N4	Start	Interval	Global Successor	Local Successor
	5	[5,6)	5	7
	6	[6,0)	7	7
	0	[0,4)	0	1

N5	Start	Interval	Global Successor	Local Successor
	6	[6,7)	7	
	7	[7,1)	7	
	1	[1,5)	1	

N6	Start	Interval	Global Successor	Local Successor
	7	[7,0)		
	0	[0,2)		
	2	[2,6)		

N7	Start	Interval	Global Successor	Local Successor
	0	[0,1)	0	1
	1	[1,3)	1	4
	3	[3,7)	3	4

L’overlay relatif à P2, pris individuellement, est quant à lui représenté par la figure 6.5

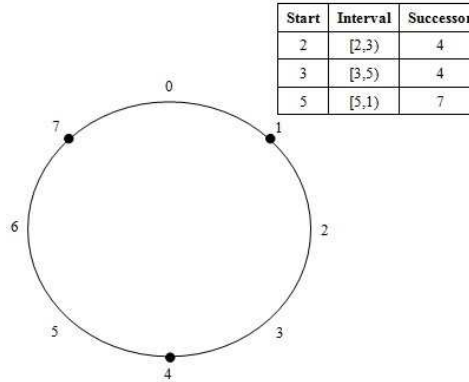


Figure 6.5 : Représentation logique de l’overlay local identifié par P2

Le tableau 6.4 représente les DHTs relatives à l’overlay local identifié par P2.

Tableau 6.4 : Vue des DHTs relatives à l'overlay local identifié par P2

N0	Start	Interval	Global Successor	Local Successor
	1	[1,2)	1	2
	2	[2,4)	2	2
	4	[4,0)	4	0

N1	Start	Interval	Global Successor	Successor
	2	[2,3)	2	4
	3	[3,5)	3	4
	5	[5,1)	5	7

N2	Start	Interval	Global Successor	Local Successor
	3	[3,4)	3	3
	4	[4,6)	4	0
	6	[6,2)	7	0

N3	Start	Interval	Global Successor	Local Successor
	4	(4,5)	4	0
	5	(5,7)	5	0
	7	(7,3)	7	0

N4	Start	Interval	Global Successor	Successor
	5	[5,6)	5	7
	6	[6,0)	7	7
	0	[0,4)	0	1

N5	Start	Interval	Global Successor	Local Successor
	6	(6,7)	7	
	7	(7,1)	7	
	1	(1,5)	1	

N6	Start	Interval	Global Successor	Local Successor
	7	(7,0)		
	0	(0,2)		
	2	(2,6)		

N7	Start	Interval	Global Successor	Successor
	0	[0,1)	0	1
	1	[1,3)	1	4
	3	[3,7)	3	4

Signalons que nous supposons le système suffisamment actif pour que tous les objets soient disponibles dans chacun des trois *overlays* considérés (P, P1 et P2). La disponibilité de l'objet au niveau local signifie que l'objet est mis en partage par l'un des nœuds représentés localement, éventuellement après une récupération du niveau global. C'est à partir de cet instant-là que nous nous positionnons pour les exemples qui suivent.

Nous avons donc que dans le cas où les pairs et objets gardent le même identifiant à tous les niveaux, alors si par exemple le nœud 2 veut récupérer l'objet 5, cela donnera d'après les tables ci-dessus :

- sans application de NETPOPPS, une recherche au niveau global uniquement et donc, un premier saut vers 4 et de là un deuxième saut vers 5 (cf. tableau 6.2 page 126) ;
- cependant avec NETPOPPS, la recherche commence au niveau local où 5 se trouve stocké sur 0, et où donc un seul saut est nécessaire pour aller de 2 à 0 (cf. tableau 6.3 page 127).

Considérons maintenant le cas où chaque pair est identifié par autant de *nodeIds* que d'*overlays* où il est présent (c.-à-d. que la fonction injective utilisée pour la dérivation du *nodeId* primaire n'est pas idempotente). Dans

ce cas, les DHTs du système sont modifiées (les tableaux précédents ne sont plus valables) et l'on doit raisonner du point de vue des nœuds physiques.

Pour retrouver alors les DHTs du système, revenons à la répartition des pairs, comme considérée plus haut (cf. page 124), où on a supposé qu'à l'observation du système, on voit que :

- dans le plan global $\{0,1,2,3,4,5,7\}$, il n'y a pas de nœud d'identifiant 6 ;
- le nœud d'identifiant 5 est présent uniquement dans P ;
- l'ensemble des nœuds $\{0,2,3\}$ du plan global se retrouve au niveau local dans P1 ;
- et l'ensemble des nœuds $\{1,4,7\}$ du plan global se retrouve au niveau local dans P 2.

De cette répartition des nœuds logiques du système, et en supposant que les *nodeIds* de l'ensemble $\{0,1,2,3,4,5,7\}$ sont des représentations logiques au plan global des nœuds physiques définis successivement par les éléments de l'ensemble $\{A,B,C,D,E,F,H\}$, on a que :

- l'ensemble des nœuds $\{A,C,D\}$ se retrouve au plan P1 ;
- et l'ensemble des nœuds $\{B,E,H\}$ se retrouvent au plan P2.

Supposons à présent que la fonction utilisée pour la dérivation des identifiants primaire est la fonction injective de permutation de 3 des éléments de l'ensemble : $h(x) : x \rightarrow (x+3) \bmod 8$.

Aux éléments de l'ensemble $\{0,1,2,3,4,5,6,7\}$, $h(x)$ fait donc correspondre les éléments de l'ensemble $\{3,4,5,6,7,0,1,2\}$ pris dans le même ordre. On a donc :

- F est représenté uniquement au plan global ; il l'est par le *nodeId* 5 ;
- $\{A,C,D\}$ seront représentés dans le plan global par $\{0,2,3\}$, pris dans l'ordre, et dans P1 par $\{3,5,6\}$, pris dans le même ordre ;
- et $\{B,E,H\}$ seront représentés dans le plan global par $\{1,4,7\}$, pris dans l'ordre, et dans P2 par $\{4,7,2\}$, pris dans le même ordre.

De tout cela découle la vue globale des DHTs au niveau du système, comme représentée par le tableau 6.5.

Tableau 6.5 : Vue globale des DHTs du système (avec une fonction injective non idempotente)

<i>N0</i>	Start	Interval	Global Successor	Local Successor
	1	[1,2)	1	
	2	[2,4)	2	
	4	[4,0)	4	

<i>N1</i>	Start	Interval	Global Successor	Local Successor
	2	[2,3)	2	
	3	[3,5)	3	
	5	[5,1)	5	

<i>N2</i>	Start	Interval	Global Successor	P2 Successor
	3	[3,4)	3	4
	4	[4,6)	4	4
	6	[6,2)	7	7

<i>N3</i>	Start	Interval	Global Successor	P1 Successor
	4	[4,5)	4	5
	5	[5,7)	5	5
	7	[7,3)	7	3

<i>N4</i>	Start	Interval	Global Successor	P2 Successor
	5	[5,6)	5	7
	6	[6,0)	7	7
	0	[0,4)	0	2

<i>N5</i>	Start	Interval	Global Successor	P1 Successor
	6	[6,7)	7	6
	7	[7,1)	7	3
	1	[1,5)	1	3

<i>N6</i>	Start	Interval	Global Successor	P1 Successor
	7	[7,0)		3
	0	[0,2)		3
	2	[2,6)		3

<i>N7</i>	Start	Interval	Global Successor	P2 Successor
	0	[0,1)	0	2
	1	[1,3)	1	2
	3	[3,7)	3	4

Pour mettre en relief les incidences positives de NETPOPPS sur l'efficacité du routage, selon la répartition des pairs (et le confinement des ressources) conformément à un modèle économique donné, nous distinguons, pour chacun des exemples qui vont suivre, trois cas différents :

- premièrement, le chemin de la requête dans le cas basique où le système fonctionne avec uniquement le réseau global (et donc avant la mise en œuvre de NETPOPPS) ;
- deuxièmement, le cas où seulement les objets gardent un même identifiant à chacun des niveaux où ils sont représentés (c'est le cas où une certaine fonction injective idempotente est appliquée pour la génération des *objectIds* uniquement) ;
- enfin, troisièmement, le cas où même les objets changent d'identifiant entre un niveau et l'autre.

Avec la fonction injective de permutation de 3, définie plus-haut, et grâce aux DHTs représentées par le tableau 6.5 (cf. page précédente), on a donc les exemples non exhaustifs suivants :

- si le pair physique C (donc 2, dans P) veut récupérer l'objet 5, on a :

- au niveau global, un premier saut a lieu vers 4 (donc pair physique D qui est dans l'entité de C suivant le modèle économique) et de là un deuxième saut vers 5 (donc pair physique E qui suivant le modèle économique n'est pas dans l'entité de D, ni de C) ;
 - cependant au niveau local, C correspond à 5, et donc s'il veut récupérer l'objet 5, l'objet 5 est bien chez lui (=> plus aucun saut : on a gagné suivant le modèle économique et on a aussi gagné en nombre de sauts logiques) ;
 - mais si l'objet change d'*objectId*, alors localement C (c.-à-d. 5) va chercher 0 ; alors il y aura un saut vers 3 (donc un seul saut, et en plus vers le pair physique A qui est dans l'entité de C suivant le modèle économique).
- si le pair physique C (donc 2, dans P) veut récupérer l'objet 6 :
- au niveau global, un saut a lieu vers 7 (donc pair physique H qui n'est pas dans l'entité de C suivant le modèle économique) ;
 - cependant au niveau local, C correspond à 5, et donc s'il veut récupérer l'objet 6, il y aura un saut vers 6 (donc aussi un seul saut, mais vers le pair physique D qui est dans l'entité de C suivant le modèle économique) ;
 - mais si l'objet change d'*objectId*, alors localement C (c.-à-d. 5) va chercher 1 ; alors il y aura un saut vers 3 (donc un seul saut, et en plus vers le pair physique A qui est dans l'entité de C suivant le modèle économique).
- si le pair physique C (donc 2, dans P) veut récupérer l'objet 0 :
- au niveau global, un saut a lieu vers 7 (donc pair physique H qui n'est pas dans l'entité de C suivant le modèle économique) et de là un deuxième saut vers 0 (donc pair physique A qui suivant le modèle économique n'est pas dans l'entité de H mais de C : donc un aller-retour vers double coût, pour revenir chez soi !) ;
 - cependant au niveau local, C correspond à 5, et donc s'il veut récupérer l'objet 0, il y aura un saut vers 3 (donc un seul saut, et en

plus vers le pair physique A qui est dans l'entité de C suivant le modèle économique) ;

- mais si l'objet change d'*objectId*, alors localement C (c.-à-d. 5) va chercher 3 ; alors il y aura un saut vers 3 (donc un seul saut, et en plus vers le pair physique A qui est dans l'entité de C suivant le modèle économique).

Les exemples non exhaustifs présentés ci-haut montrent bien une amélioration du diamètre (nombre de sauts pour une requête) conforme au modèle 'économique' considéré. Ils nous montrent aussi que l'utilisation d'une fonction idempotente pour la dérivation de l'*objectId* primaire peut introduire encore des gains sur le chemin d'une requête (en tout cas pas de perte). Ces exemples nous montrent aussi que notre contribution respecte la localisation du chemin des requêtes conformément au confinement des ressources selon le modèle économique considéré.

CONCLUSION DU CHAPITRE –

Ce chapitre a établi une analyse comparative et applicative de nos deux contributions. Il a en plus expliqué un exemple illustré de leur incidence positive sur l'efficacité du routage P2P.