

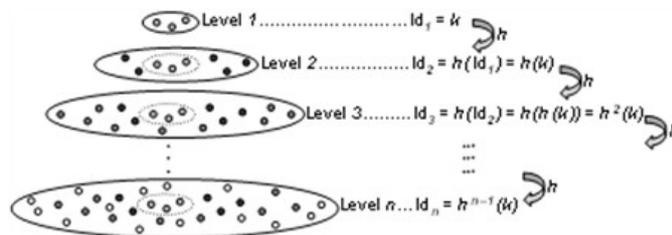
**DÉFINITION, CONCEPTION, SPÉCIFICATION ET ANALYSE  
D'UNE SOLUTION PAIR-À-PAIR ORIENTÉE  
OPÉRATEUR DE RÉSEAUX NETPOPPS A  
NETWORK PROVIDER ORIENTED PEER-TO-PEER  
SYSTEM**

*Ce chapitre définit, spécifie et analyse un système P2P orienté opérateur de réseau. Cette solution, baptisée NETPOPPS (NETwork Provider Oriented P2P System), adapte les flux P2P à la topologie réseau tout en simplifiant la gestion des identifiants des différents pairs et objets. Elle utilise pour cela la technique de dérivation de clés.*

*Le chapitre commence par la présentation du principe de la dérivation de clés.*

**5.1 - Principe de la dérivation de clés**

La figure 5.1 illustre le principe de la dérivation de clés. La fonction  $h$  est une fonction injective quelconque. Dans le cas d'un système P2P,  $h$  sera la fonction de hachage utilisée par la DHT du protocole P2P mis en œuvre au niveau global (non représenté sur la figure).



**Figure 5.1 :** Principe de la dérivation de clé pour la gestion de clés

Le principe de la dérivation permet d'obtenir à partir d'une même clé plusieurs clés en relation mathématique simple entre elles et avec la clé d'origine. Dans l'exemple représenté à la figure 5.1, ce mécanisme

s'applique à chaque niveau de la hiérarchie, à partir du niveau supérieur. Au niveau 1 (*Level 1*) chaque élément a un identifiant  $Id_1 = k$ . C'est la clé. Elle est dérivée (par application de  $h$ ) pour donner l'identifiant du même élément au niveau suivant. Ainsi cet élément est identifié au niveau 2 (*Level 2*) par  $Id_2 = h(Id_1)$ , au niveau 3 (*Level 3*) par  $Id_3 = h(Id_2) = h(h(Id_1)) = h^2(Id_1)$ , et ainsi de suite. Cette relation mathématique peut être vérifiée entre n'importe quels identifiants d'un même élément à deux niveaux strictement successifs. En d'autres termes, à tout niveau  $i$  (*Level i*), un élément est identifié par  $Id_i = h(Id_{i-1}) = h^{i-1}(Id_1)$ . Et ainsi, tout élément peut déduire son identifiant  $Id_j$  au niveau  $j > i$  en dérivant  $(j-i)$  fois son identifiant  $Id_i$  du niveau  $i$ , c.-à-d.  $Id_j = h^{j-i}(Id_i)$ .

Le principe de la dérivation de clés a déjà été utilisé dans le cadre de la sécurisation des communications privées de groupes hiérarchiques nécessitant une confidentialité particulière dans chacun des sens ascendant et descendant de la hiérarchie [HBB<sup>+</sup>05]. Nous l'utilisons ici à des fins de routage efficace transparent, indépendamment de toute exigence de sécurité.

## 5.2 - Architecture de NETPOPPS

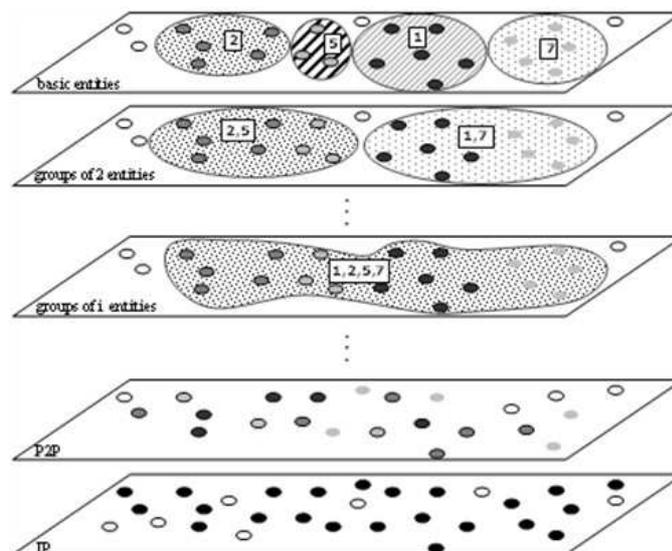
Le réseau Internet est composé de dizaines de milliers d'ASs reliés entre eux suivant des accords économiques signés entre les opérateurs qui les gèrent [Meu07]. Un réseau P2P interconnecte donc des pairs de différents ASs. Il peut alors se décomposer en plusieurs couches *overlays* secondaires, disposées au-dessus du réseau global tel que chacune soit conforme à un relevé précis (c.-à-d. une politique, une priorité, une métrique) des accords inter-opérateurs. Pour la présentation de l'architecture de NETPOPPS, nous considérons la métrique distance, exprimée en nombre de sauts d'un AS à l'autre. La priorité donnée à un AS parmi d'autres pour y aller au prochain saut résulte des accords inter-opérateurs négociés par l'opérateur gérant l'AS de départ.

NETPOPPS est cependant une architecture générale non spécifique à un opérateur de réseau donné, tel un FAI. Elle peut être adoptée par exemple

par des universités ou des compagnies multi-sites ayant des politiques dotées de priorités pour la gestion des flux entre leurs différents sites, notamment dans le cadre de certains services P2P.

Pour cela, dans la suite, nous désignons l'opérateur de réseau par NP (pour *Network Provider*) et nous remplaçons le terme AS par « entité de base » ou BE (pour *Basic Entity*). Ainsi un NP peut gérer une ou plusieurs BEs. Dans ce cas, il pourra y avoir priorités et politiques intra-NP à l'égard de ses BEs ; mais dans la suite, nous ne parlerons que d'accords inter-opérateurs.

La fusion de deux BEs est possible virtuellement, dès lors que les accords signés entre leurs NPs autorisent des échanges entre un pair de l'une et un pair de l'autre. La nouvelle entité virtuelle est dite une entité intermédiaire (entre la BE et l'ensemble de tous les pairs actifs) ou IE (pour *Intermediate Entity*). Deux IEs peuvent à nouveau fusionner en une nouvelle IE, et ainsi de suite, tant que la nouvelle entité ne s'est pas confondue avec l'*overlay* P2P global existant. Chaque IE étant la fusion de plusieurs BEs, elle sera gérée indépendamment par autant de NPs. La concordance de ces gestions indépendantes est garantie par la concordance des accords inter-opérateurs.



**Figure 5.2 :** Vue d'ensemble d'un système NETPOPPS

La définition d'entités (BE ou IE) garantit que le chemin de connexion P2P, émergeant d'un pair quelconque de l'entité et aboutissant à un autre quelconques de ses pairs, reste entièrement dans la même entité.

La figure 5.2 illustre une vue d'ensemble du système. Au-dessus de l'*overlay* P2P initial,  $l$  couches *overlays* secondaires sont organisées comme suit. Au niveau le plus élevé, se trouve les *overlays* formés chacun par les pairs d'une même BE. C'est le niveau local. Les niveaux qui le séparent du niveau P2P global sont dits intermédiaires et regroupent des *overlays* formés chacun par les pairs d'une même IE. Chaque niveau intermédiaire est caractérisé par le diamètre de ses IEs. Il s'agit du nombre de BEs qui forment les IEs en conformité avec les accords inter-opérateurs (ou politique des NPs). Ce nombre va croissant en passant d'un niveau secondaire à l'autre, allant du niveau le plus élevé au niveau le plus bas (l'*overlay* initial global). C'est l'algorithme de routage (e.g. Chord [SML<sup>+</sup>01], Pastry [RD01], etc.) mis en œuvre au niveau global qui sera repris dans chacun des *overlays* intermédiaires.

Chaque pair est présent à tous les niveaux. Et chaque NP est responsable, à chaque niveau, de toutes les IEs basées sur les BEs qu'il gère. Une IE peut donc être gérée indépendamment par deux NPs différents. La cohérence des accords entre NPs, sur la base desquels les IEs sont formées, garantit la cohérence parfaite de la gestion assurée par chaque NP.

### 5.2.1 - Identification des entités

Il est essentiel que des pairs provenant de deux BEs différentes et se retrouvant dans une même IE reconnaissent qu'il s'agit là de la même entité d'appartenance. Pour cela, nous supposons que chaque entité a un identifiant global unique, que nous appelons « vecteur identifiant » ou tout simplement : vecteur.

L'unicité globale des identifiants de chaque BE est tout à fait justifiée. Dans le cas où une BE se confond avec un AS, le vecteur de la BE sera l'ASN (*AS Number*).

Quant à l'identifiant d'une IE, il se présente sous la forme d'un vecteur (d'où la désignation généralisée) qui liste par ordre croissant tous les identifiants des différentes BEs qui forment la IE.

La figure 5.2 affiche le vecteur de chaque entité représentée. L'exemple illustré se focalise sur quatre BEs identifiées par les vecteurs "1", "2", "5", et "7". Il suppose les déclarations suivantes émanant de l'ensemble des accords signés par les différents NPs gérants de ces BEs :

- les connections P2P sortantes de la BE "1" doivent essayer de se satisfaire dans la BE "7" en première priorité (et vice-versa), avant d'essayer d'aboutir à des pairs des BEs "2" ou "5" en priorité  $i$  ;
- les connections P2P sortantes de la BE "2" doivent essayer de se satisfaire dans la BE "5" en première priorité (et vice-versa), avant d'essayer d'aboutir à des pairs des BEs "1" ou "7" en priorité  $i$ .

Par conséquent, au premier plan suivant le niveau local, les BEs de vecteur "1" et "7" fusionnent virtuellement en une même IE de vecteur "1,7", et les BEs de vecteur "2" et "5" fusionnent virtuellement en une même IE de vecteur "2,5". De même, au niveau intermédiaire de rang  $i$  à partir du plan local, les quatre BEs représentées fusionnent virtuellement en une même IE de vecteur "1,2,5,7".

Une BE peut donner aussi des priorités égales à deux autres BEs qui cependant ne se donnent pas la même priorité. Ceci peut notamment advenir dans le cas où les NPs sont des FAIs. En voici un exemple : les BEs "2", "5" et "6" (non représenté sur la figure) qui se regroupent en une IE "2,5" et une IE "5,6" au premier niveau intermédiaire et en la IE "2,5,6" au niveau suivant. Les pairs de la BE "5" seront donc représentés au premier niveau intermédiaire dans deux IEs différentes, mais fusionnent à nouveau en une

représentation unique au niveau qui suit. Dans la suite, nous référençons ce cas par la dénomination « représentation double ».

Chaque vecteur caractérise aussi le réseau *overlay* formé par les pairs d'une entité, ainsi que la DHT de l'espace d'identification correspondant. Cette nouvelle DHT référence donc une partie des éléments (pairs et objets) de l'*overlay* global mais regroupés différemment. Elle est alors virtuelle par rapport à la DHT de base du plan P2P global. Elle sera appelée VDHT (pour *Virtual DHT*).

Afin de normaliser les vecteurs, leur taille peut être fixée à la longueur maximale possible. Dans ce cas, les vecteurs de longueur inférieure peuvent être complétés par des zéros à gauche. Par exemple, le vecteur "5" se transformera en un vecteur "0,...,0,5".

Par ailleurs, pour des soucis d'homogénéisation du système, le réseau P2P global peut se voir affecté un vecteur nul (c.-à-d. "0", ou "0,...,0" selon le cas).

### 5.2.2 - Identification des pairs

Chaque pair a un *nodeId* dans chaque *overlay* où il existe, et donc autant de *nodeIds* que le nombre total d'*overlays* auxquels il participe. Ceci est analogue à un terminal ayant plusieurs interfaces réseaux et donc autant d'adresses IP : une par réseau. Dans un souci de simplicité, nous n'exigeons pas d'unicité entre les différents espaces d'identification.

Par ailleurs, en observant de près l'architecture de NETPOPPS, nous remarquons que chaque BE forme avec les IEs où ses pairs se trouvent un sous-système hiérarchique géré par le NP propriétaire de la BE en question (cf. fig. 5.2 p. 99). (Dans le cas d'une « représentation double », la BE chapeaute deux sous-systèmes hiérarchiques.)

Les niveaux *overlays* du système constituent alors un ensemble de plusieurs sous-systèmes hiérarchiques gérés par les NPs. Chaque NP gère autant de sous-systèmes qu'il gère de BEs. Chaque sous-

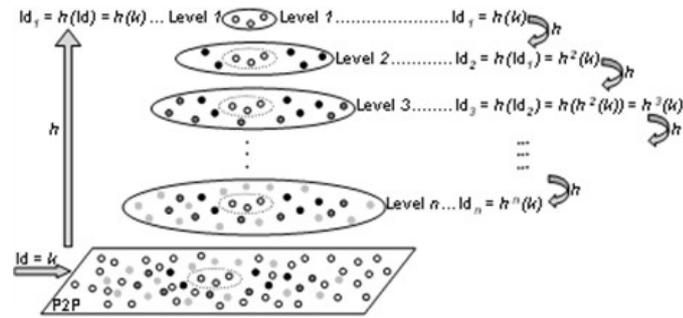
système est formé d'une BE au sommet d'autant d'IEs qu'il y a de niveaux intermédiaires dans le système.

Donc pour permettre une coopération effective du trafic P2P avec les politiques des NPs en matière de gestion du trafic, un mécanisme d'autogestion du trafic P2P serait nécessaire à l'intérieur de chaque sous-hiérarchie du système.

Par souci d'indépendance par rapport au protocole P2P, et comme nous abordons uniquement les systèmes utilisant des DHTs, il est utile d'introduire un mécanisme de gestion des DHTs. Par ailleurs, une hiérarchie isolée ressemble bien à l'illustration donnée par la figure 5.1 (cf. p. 97), où  $n$  correspondrait au nombre d'*overlays* secondaires au dessus de l'*overlay* P2P global dans NETPOPPS.

Pour cela, le mécanisme de gestion des DHTs qui est introduit dans le système, dans chaque sous-hiérarchie, est basé sur le principe de dérivation de clés.

Par conséquent, les différents *nodeIds* de chaque pair seront en relation mathématique simple, comme suit :  $Id_i = h(Id_{i-1}) = h^{i-1}(Id_1)$  et  $Id_j = h^{j-i}(Id_i)$ , où  $j$  et  $i$  sont des niveaux intermédiaires quelconques avec  $j > i$  et  $Id_1$  est le *nodeId* du pair dans l'*overlay* du BE (cf. paragraphe 5.1 p. 97). Cependant, comme NETPOPPS s'intègre à un système P2P existant, chaque pair va entrer dans le système à partir de l'*overlay* global, où il va alors acquérir son premier *nodeId*. Soit  $k$  ce *nodeId* que nous désignerons par « *nodeId* primaire ». Pour lier  $k$  à  $Id_1$  à la manière des autres *nodeIds* du pair, une dérivation supplémentaire est insérée dans le processus, tel que  $Id_1 = h(k)$ . Il en découle pour tout niveau intermédiaire  $i$ ,  $Id_i = h^i(k)$ . (On a toujours  $Id_j = h^{j-i}(Id_i)$  pour  $j > i$ ) Cette variante du principe de la dérivation de clés est illustrée à la figure 5.3.



**Figure 5.3 :** Calcul des différents identifiants d'un même nœud

Dans le cas d'une « représentation double », un pair aura à un niveau donné, le même *nodeId* dans chacune des hiérarchies.

### 5.2.3 - Identification des objets

Comme un pair participe à plusieurs *overlays*, il partage alors ses ressources dans chacun de ces *overlays*.

#### 5.2.3.1 - Approche générale

Une même ressource partagée dans différents *overlays* va avoir autant d'identifiants (*objectIds*) que le pair qui la partage. Ceci revient à donner plusieurs pseudonymes à un même objet. De même que pour les *nodeIds*, nous n'exigeons pas d'unicité entre les différents espaces d'identification.

Les différents *objectIds* sont en relation mathématique conforme au principe de la dérivation de clés adapté (tout comme les *nodeIds*). Un objet acquiert donc son premier identifiant dans l'*overlay* global : c'est l'« *objectId* primaire ». Et pour chaque objet, on a à tout niveau  $i$  intermédiaire :  $Id_i = h(Id_{i-1}) = h^{i-1}(Id_1) = h^i(Id_{primaire})$ .

Cependant avec NETPOPPS, un pair ne peut partager ses ressources qu'une fois qu'il aura adhéré complètement au système, c.-à-d. lorsqu'il se sera inséré dans chacun des différents *overlays* auxquels il doit appartenir.

### 5.2.3.2 - Approche particulière

Au cas où tous les espaces d'identification du système sont de même taille, et comme ils sont de même nature sans aucune caractéristique distinctive qui impacte le calcul des différents identifiants, un objet peut être partagé avec le même *objectId* dans chaque *overlay*, d'autant plus que les objets sont passifs et gérés par les pairs.

Cette alternative permet à tout objet d'avoir un *objectId* unique dans tout le système. Ceci revient à utiliser pour tout le système (y compris dans l'*overlay* global) une DHT utilisant une fonction de hachage idempotente, c.-à-d. qui appliquée une, deux, ou même  $n$  fois à elle-même donne toujours le même résultat.

Ceci dit, une fonction de hachage idempotente pourrait bien aussi être utilisée uniquement pour les *overlays* secondaires. Ce qui donnera à chaque objet un *objectId* primaire dans l'*overlay* initial (global) et un seul et même autre *objectId* à tous les autres niveaux.

## 5.3 - Mécanisme de routage dans NETPOPPS

Le mécanisme de routage est le même dans chaque *overlay*. Il s'agit de celui mis en œuvre au niveau P2P global (e.g. Chord [SML<sup>+</sup>01], cf. paragraphe 1.4.2 p. 44). Une différence s'installe toutefois d'un *overlay* à l'autre : les *nodeIds* utilisés, les *objectIds* (s'ils sont différents), ainsi que la DHT (ou VDHT) consultée.

Quand un pair veut rechercher un objet, il commence par calculer son *objectId* primaire qu'il garde en antémémoire le temps de la requête pour le cas où il aura à chercher dans l'*overlay* global. Puis il dérive l'*objectId* primaire et utilise cette nouvelle clé dans sa requête qu'il lance dans l'*overlay* local où il réside en s'identifiant par son *nodeId* correspondant et

en utilisant la VDHT correspondante. Ainsi la requête n'atteint que des pairs de son BE.

Si la requête n'aboutit pas, elle est alors relancée dans l'*overlay* de l'IE suivant au niveau strictement inférieur de la hiérarchie, avec les DHT, *nodeId* et éventuellement *objectId* correspondants. Et ainsi de suite, jusqu'à satisfaction de la requête.

Ce n'est qu'en dernier lieu, qu'une requête est lancée dans l'*overlay* global, si elle n'a pas encore abouti. À ce niveau, ce sont les services de la DHT globale qui sont utilisés par le *nodeId* primaire pour trouver l'objet identifié, selon le cas (approche particulière ou générale des espaces d'identification), par le même *objectId* ou par l'*objectId* primaire gardé en antémémoire. Il est toutefois rare d'atteindre l'*overlay* global si le système est bien peuplé et la charge y est bien équilibrée.

Il est à noter que seul le pair initiateur d'une requête peut relancer (à un niveau inférieur) une requête non aboutie. Aussi, un pair reconnaît la VDHT qu'il doit utiliser grâce au *nodeId* par lequel il a été sollicité.

Dans le cas d'une « représentation double » le pair va se retrouver avec deux VDHTs à un même niveau intermédiaire. Alors lorsqu'il sera sollicité dans l'un des *overlays* où il se trouve, et afin d'éviter tout conflit, il va demander à celui qui le sollicite de lui donner le vecteur de son *overlay* (que l'autre reconnaît de par son *nodeId* ou la VDHT qu'il utilise). Si par contre, un des pairs de la BE intégrée dans deux IEs à un même niveau initie une requête, le mécanisme de routage se poursuit à l'identique et simultanément dans les deux hiérarchies que la BE chapeaute.

Enfin, quand une ressource est trouvée dans un *overlay* non local (c.-à-d. global ou intermédiaire), le pair qui la récupère la remet en partage dans son propre *overlay* local, tout comme il y partagerait une nouvelle ressource.

#### 5.4 - Mouvement des pairs dans NETPOPPS

Les mouvements d'arrivée et de départ d'un pair sont gérés de façon identique dans tous les *overlays* du système et conformément au protocole

de routage mis en œuvre dans l'*overlay* global. Cependant après s'être joint à l'*overlay* global et avant de pouvoir se joindre à n'importe lequel des *overlays* secondaires, un pair  $n$  doit effectuer certaines opérations. En effet, il doit identifier ces *overlays* et connaître au moins un autre pair  $n_1$  actif dans chacun d'eux. Ce qui l'amènerait à devoir connaître autant d'autres pairs distincts que d'*overlays* secondaires où il cherche à s'insérer. Mais étant orienté opérateur de réseaux, NETPOPPS propose de ramener tous ces différents pairs à un seul nœud géré par le NP. Ce nœud sera un nœud de référence pour les nouveaux pairs lors de leur adhésion au système. Il sera appelé nœud de contrôle (ou *CN* pour Control Node).

#### 5.4.1 - Le nœud de contrôle

Chaque NP insère dans le système un CN par BE qu'il gère. Le CN se retrouve alors dans toutes les autres entités (IE) de la sous-hiérarchie chapeauté par la BE. Par souci de robustesse, un nœud miroir peut aussi exister pour chaque CN.

Chaque NP est libre de définir les fonctionnalités qu'il souhaite déployer sur chacun des CNs qu'il gère. Un CN peut donc avoir plusieurs rôles ; en particulier, il détient un profil sauvegardé et géré par le NP. Chaque NP est aussi libre de définir et sauvegarder dans ce profil les données qui l'intéressent. En particulier, le profil mémorise les identifiants des entités (c.-à-d. les vecteurs de la BE et des IEs) dans lesquelles le CN réside. L'amplitude et diversification des autres fonctionnalités, ainsi que les détails du restant des données du profil n'affectent ni le mécanisme de routage P2P ni les opérations des pairs. Nous ne les abordons donc pas.

Le CN est présent dans chaque *overlay* correspondant à une entité où il réside. Il a donc plusieurs identifiants. Comme tout pair, il entre dans le système à partir de l'*overlay* global, où il acquiert son *nodeId* primaire,  $Id_{CN,ref}$ . Et ses différents identifiants sont aussi en relation mathématique conforme au principe de la dérivation de clé modifié (cf. figure 5.3 p. 103). Quand le niveau de rang 1 est celui de

la BE, on a donc pour tous niveaux intermédiaires de rang  $i$  et de rang  $j > i$ ,  $Id_{CN,i} = h^i(Id_{CN,ref})$  et  $Id_{CN,j} = h^{j-i}(Id_{CN,i})$ .

Ainsi lorsque le CN a acquis son  $Id_{CN,ref}$  il calcule ses différents identifiants. Et à chaque nouvel identifiant calculé, il initie l'*overlay* correspondant conformément au profil. Quand le CN a réussi les initialisations des différents *overlays* le concernant, il génère une table de référence (ou *RT* pour *Reference Table*).

#### 5.4.2 - La table de référence

La RT est une table qui récapitule tous les vecteurs des différentes entités d'appartenance du CN. Dans le cas d'une « représentation double », la RT sera une structure matricielle dont chaque ligne sera la RT relative à l'une des hiérarchies concernées.

La RT permet donc aux nœuds du même BE que le CN qui arrivent dans le système de connaître les différents IEs de la hiérarchie. Quand le CN crée la RT, il la met donc en partage dans l'*overlay* global. Elle y est gérée comme un objet, et par souci de robustesse, peut être répliquée dans ce même *overlay*, en particulier, elle reste stockée sur le CN.

Comme chaque BE a un vecteur unique, qu'il y a un CN par BE et une RT par CN, la RT est unique pour chaque BE. Chaque RT peut donc être identifiée de façon unique dans le système par  $h(\text{vector\_of\_BE})$ . Il y a donc dans le système autant de RTs que de CNs et que de BEs.

La structure de la RT diffère cependant selon le degré d'interaction entre le NP et les pairs se joignant au système. Comme la RT est générée par le CN qui est géré par le NP, le choix de la mise en œuvre de l'une ou l'autre des structures de la RT revient au NP. On distingue entre un système passif et un système actif. Le choix de cette dénomination s'éclaircira à la présentation du mécanisme d'adhésion d'un nouveau pair au système (cf. paragraphe 5.4.3 p. 110).

#### 5.4.2.1- Système passif

La structure de la RT en mode dit passif est donnée par le tableau 5.1. C'est une table à deux entrées et  $(N+1)$  colonnes, où  $N$  est le nombre de niveaux intermédiaires dans le système, et donc de IEs dans la hiérarchie considérée.

**Tableau 5.1 : Structure de la RT en mode passif**

<i>vector of the BE</i>	<i>vector of the 1<sup>st</sup> IE</i>	...	<i>vector of the N<sup>th</sup> IE</i>
<i>local ID of CN</i>	<i>CN's ID in the 1<sup>st</sup> IE</i>	...	<i>CN's ID in the N<sup>th</sup> IE</i>

Les différents champs de cette RT sont les suivants :

- à la première ligne de la première colonne se trouve le vecteur de la BE du CN (*vector of the BE*) ;
- à la deuxième ligne de la première colonne se trouve l'identifiant local du CN, c.-à-d. le *nodeId* du CN dans la BE (*local ID of CN = Id<sub>CN,1</sub>*) ;
- pareillement, les autres colonnes donnent à la première ligne le vecteur d'une IE et à la deuxième ligne le *nodeId* du CN dans cette IE. Les colonnes se suivent dans l'ordre de superposition des IEs, en allant de la première IE strictement en dessous de la BE, pour la deuxième colonne, à la IE strictement au dessus du plan P2P global, pour la dernière colonne.

#### 5.4.2.2- Système actif

La structure de la RT en mode dit actif est donnée par le tableau 5.2. C'est une table à une seule entrée et  $(N+2)$  colonnes, où  $N$  est le nombre de niveaux intermédiaires dans le système, et donc de IEs dans la hiérarchie considérée.

**Tableau 5.2 : Structure de la RT en mode actif**

<i>Vector of the BE</i>	<i>primary ID of the CN</i>	<i>vector of the 1<sup>st</sup> IE</i>	...	<i>Vector of the N<sup>th</sup> IE</i>
-----------------------------	---------------------------------	--	-----	--

Les différents champs de cette RT sont les suivants :

- le premier champ donne le vecteur de la BE du CN ;
- le deuxième champ donne le *nodeId* primaire du CN,  $Id_{CN,ref}$  ;
- les autres champs contiennent successivement les vecteurs de toutes les IEs, en allant de la première IE strictement en dessous de la BE, pour la troisième colonne, à la IE strictement au dessus du plan P2P global, pour la dernière colonne.

#### 5.4.3 - Arrivée d'un pair

Un nouveau pair arrive dans le système en se joignant à l'*overlay* global, où il acquiert son *nodeId* primaire. Pour pouvoir se joindre correctement aux bons *overlays* secondaires, il doit maintenant connaître les différents identifiants du CN de sa BE d'appartenance. Pour cela il doit retrouver la RT qui lui donnera les informations utiles.

Par un mécanisme externe à NETPOPPS, chaque pair peut connaître sa BE d'appartenance. Il lui faudra pour cela se servir des bases d'information du système (e.g. des tables BGP [RFC1771] pour avoir les numéros des ASs d'appartenance). Ce type de mécanisme n'est pas discuté dans la suite. NETPOPPS s'intéresse principalement aux mécanismes de routage et aux mouvements des pairs.

Quand le nouveau pair reconnaît sa BE d'appartenance, il connaît le vecteur de celle-ci. Il calcule alors l'identifiant de la BE dans l'*overlay* global, c.-à-d.  $h(\text{vector\_of\_BE})$ . Cet identifiant étant aussi celui de la RT (dans l'*overlay* global), le nouveau pair initie une requête *reference\_table\_request()* au niveau global pour

recupérer la RT. Deux cas sont alors possibles suivant le degré d'interaction entre le NP et les pairs se joignant au système.

(Dans le cas d'une « représentation double », la RT récupérée sera une matrice dont chaque ligne est une RT pour l'une des hiérarchies. Le mécanisme d'adhésion se poursuit alors simultanément dans chaque hiérarchie comme décrit ci-dessous)

#### 5.4.3.1 - Systeme passif

Dans le cas passif, lorsque le nouveau pair récupère la RT, il connaît tous les vecteurs des différentes entités où il réside logiquement, ainsi que les différents *nodeIds* du CN de sa BE dans chacun de ces *overlays*.

Le nouveau pair calcule alors ses différents identifiants selon le principe adapté de la dérivation de clé (cf. paragraphe 5.2.2 p. 102). Et à chaque nouvel identifiant calculé, il fait la correspondance avec les données fournies par la RT en avançant d'une colonne. Ainsi le nouvel identifiant du nouveau pair correspond à la nouvelle entité identifiée. Et grâce au *nodeId* correspondant du CN, le nouveau pair se joint à l'*overlay* correspondant.

Par exemple, si le protocole mis en œuvre au niveau global est Chord ([SML<sup>+</sup>01], cf. paragraphe 1.4.2 p. 44), et que le *nodeId* primaire du nouveau pair est  $n$  et celui du CN de sa BE  $Id_{CN}$ , l'opération d'adhésion à un niveau intermédiaire  $i$  s'exécute à travers la fonction  $Id_{i,join}(Id_{CN,i}) = [h^i(n)].join[h^i(Id_{CN})]$ .

#### 5.4.3.2 - Systeme actif

Dans le cas actif, lorsque le nouveau pair récupère la RT, il prend connaissance de tous les vecteurs des différentes entités où il réside logiquement, ainsi que de  $Id_{CN,ref}$ , le *nodeId* primaire du CN de sa BE.

Il y a là deux mises en œuvre possibles de NETPOPPS selon que le NP ait ou pas un plan d'action à opérer sur les nœuds se joignant aux différents *overlays* qu'il gère.

(Dans le cas d'une « représentation double » la mise en œuvre sera pareille pour toutes les hiérarchies issues de la BE.)

Dans les deux cas, une fois que le nouveau pair connaît les différents *nodeIds* du CN de sa BE dans chacun des *overlays* secondaires, le mécanisme de son adhésion à ces différents *overlays* se poursuit comme dans le cas d'un système passif.

a) *Le NP a défini un plan d'action*

Dans l'affirmative, le nouveau pair envoie au CN dans le plan global une requête d'intention d'adhésion à chacun des *overlays* secondaires dont il a pris connaissance du vecteur. Le CN exécute alors le plan d'action prévu par le NP. Il peut par exemple opérer un contrôle d'accès en vérifiant les politiques d'accès. Il peut aussi appliquer un mécanisme propriétaire quelconque (e.g. pour un service payant, pour une récupération de données à des fins statistiques, etc.). Une fois l'opération réussie, le CN envoie son *nodeId* dans l'*overlay* secondaire en question.

L'intention d'adhésion est exprimée par des requêtes indépendantes pour chaque *overlay*. Mais elle peut bien être exprimée aussi par une seule requête incluant tous les vecteurs. Selon le cas, le CN renverra ses *nodeIds* secondaires individuellement, en réponse à chaque requête concernée, ou en bloc, si la requête est globale, mais dans le même ordre de classement des vecteurs dans la RT.

b) *Le NP n'a pas défini de plan d'action*

Si le NP n'entend pas appliquer de mécanisme particulier à ses nœuds entrant dans le système P2P, il peut toujours mettre en œuvre NETPOPPS de façon à ce que le nouveau pair calcule lui-même les différents identifiants du CN par dérivations successives de  $Id_{CN,ref}$  conformément au principe de la dérivation de clé.

#### **5.4.4 - Disparition d'un pair**

La disparition d'un pair du système (départ explicite, départ silencieux, ou panne) est généralement remarquée quasi-simultanément dans tous les *overlays* auxquels il appartenait. Elle est traitée de façon identique mais indépendante dans chaque *overlay*, et conformément au protocole mis en œuvre au niveau global.

### **5.5 - Analyse de NETPOPPS**

L'application du principe de la dérivation de clé aux identifiants d'un réseau P2P (principalement les *nodeIds*) permet la conception d'un système orienté opérateur de réseaux où le trafic P2P coopère avec les accords économiques de routage inter-opérateurs.

Pour la dérivation des identifiants, NETPOPPS peut utiliser des fonctions injectives différentes de la fonction de hachage utilisée au niveau global. Il peut s'agir d'une même fonction à tous les niveaux secondaires ; mais il peut aussi s'agir de plusieurs fonctions à condition qu'au niveau strictement supérieur au niveau global toutes les BEs d'une même IE utilisent la même fonction. Dans les deux cas, c'est le CN qui communique (par le profil) la fonction de dérivation à tout nouveau pair qui le contacte lors de son arrivée au niveau global.

Ce mécanisme de gestion des identifiants (par le principe de la dérivation de clé) permet de structurer un réseau P2P en des couches secondaires virtuelles conformes à un critère ou l'autre des accords inter-

opérateurs. Ceci permet un cloisonnement de ressources conforme à des critères économiques.

Le réseau P2P initial étant formé de pairs d'appartenance assez variée se transforme ainsi en une collection de sous-systèmes hiérarchiques aux niveaux uniformes, tel que chaque hiérarchie est gérée par un opérateur.

Chaque pair est ainsi logiquement présent à chaque niveau à travers des *nodeIds* différents, donc des pairs virtuels. De même, chaque objet est présent dans chaque *overlay* où réside le pair qui le stocke. Cette présence se manifeste à travers des *objectIds* différents. Néanmoins dans certains cas, un objet peut garder un même *objectId* dans tout le système puisque les objets sont passifs. Ce qui, en toute évidence, allège le système et contribue à diminuer les temps de latence et accélérer les procédures de recherche et de récupération des objets.

Chaque pair (ou objet) acquiert son premier identifiant (l'identifiant primaire) au niveau global. Comme la fonction de dérivation est injective, on ne trouve donc pas dans le plan local le même *nodeId* (ou *objectId*) dans deux BEs distinctes. De même, il n'y a pas deux *nodeIds* (ou *objectIds*) identiques dans deux IEs distinctes d'un même plan, à moins que le pair soit représenté deux fois.

Ainsi, si dans une des IEs, un *nodeId* (ou *objectId*) existe déjà, c'est qu'il s'agit du même pair (ou objet). Les représentations virtuelles fusionnent alors (avec le regroupement des objets à la charge de chaque représentation sur le même pair). C'est le cas des représentations doubles.

Chacune des couches secondaires créées virtuellement (par le principe de la dérivation de clé) a son propre espace d'identification et donc une DHT correspondante qui est virtuelle et dite VDHT. Celle-ci permet une gestion et une recherche circonscrites des données.

En effet, la recherche d'une ressource ou d'un élément de données dans NETPOPPS commence toujours dans l'*overlay* local (de la BE) en utilisant les services de la VDHT correspondante. Et en cas d'échec, la requête est relancée dans l'entité (IE) strictement suivante hiérarchiquement. Or, chaque entité est définie conformément à des critères d'intérêt pour le

NP et de sorte qu'un chemin entre deux de ses pairs ne passe pas par l'extérieur. De plus, chacune a un vecteur unique qui caractérise aussi sa DHT. Et chaque VDHT ne référence que les identifiants des nœuds et objets dans l'entité. D'ailleurs chaque pair a un identifiant pour chaque *overlay* où il réside. (Le routage dans le cas d'une « représentation double » a été présenté au paragraphe 5.3 p. 104.) L'identifiant de chaque objet est aussi unique dans chaque *overlay* : si ceci est clair dans l'approche générale, il l'est tout aussi dans l'approche particulière qui peut être un cas particulier idempotent du cas général (cf. paragraphe 5.2.3.1 p. 104). En tout cas, un *objectId* ne peut être référencé dans une VDHT que si l'objet est bien dans l'*overlay* correspondant, mais aussi le pair qui a la charge de l'objet.

Ainsi, toute requête qui aboutit dans l'*overlay* où elle a été initiée est garante de l'existence de l'objet dans cet *overlay*. Il en découle donc que le pair initiateur de la requête peut récupérer l'objet d'un pair répondant à des critères inter-opérateurs identiques et ce, à travers un chemin conforme à ce(s) critère(s) ne passant pas par une entité (IE) plus large.

Quant au CN, c'est un nœud de contrôle appartenant au NP. Donc à moins qu'il y ait une panne, il ne quittera pas le système : c'est un nœud quasi-statique.

Le CN n'est pas un super-pair : il ne joue pas de rôle central par rapport aux pairs de son entité, et il n'a aucune priorité dans le routage. Il sert de pont entre le NP et ceux des nœuds du même BE joignant le système P2P global :

- d'une part, il permet au NP de cadrer le trafic P2P et de le libérer graduellement (d'où les sous-systèmes hiérarchiques) conformément aux accords inter-opérateurs. Le CN permet aussi au NP d'exercer un contrôle sur ses pairs ou de leur offrir l'accès à un service propriétaire (cf. paragraphe 5.4.3.2-a p. 112) ;
- d'autre part, le CN permet aux pairs d'une même BE de se former correctement en réseaux *overlays* secondaires correspondants à la BE et à chacune des IEs englobant cette BE. Pour cela, il se manifeste auprès

des pairs se joignant à l'*overlay* global, à travers la RT qu'il génère et met à disposition dans le réseau P2P initial (c.-à-d. global).

Comme chaque IE est formée de plusieurs BEs et peut être gérée par plusieurs NPs, alors il y réside autant de CNs différents que ses BEs de base, et ces CNs appartiennent à au plus autant de NPs différents. Tous ces CNs offrent alors les mêmes fonctionnalités et contiennent les mêmes données vitales à l'unité de l'IE. Ces exactitudes proviennent de la concordance des accords inter-opérateurs. Toutefois, cela n'empêche pas chaque CN d'offrir des fonctionnalités à seule destination des propres pairs de sa BE, ni de stocker des données concernant uniquement ces pairs-ci.

Par ailleurs, la définition et la négociation des accords inter-opérateurs peuvent être une charge quasi-permanente pour un NP, et le sont particulièrement si le NP est un FAI. Si les termes de ces accords viennent donc à être modifiés, la mise à jour de certaines données du profil d'un CN peut s'avérer nécessaire. C'est le cas où une modification des accords inter-opérateurs porterait sur la modification de la composition des IEs ou la création de nouvelles IEs. Dans les deux cas, il s'agit d'une redistribution des associations entre les différentes BEs au niveau hiérarchique des IEs concernées. Mais une modification de IEs existantes ne concernera pas toutes les BEs qui la composent. Tous les pairs de cette IE ne seront donc pas concernés par ce remaniement. En tout cas, la mise à jour du profil sur le CN est à la charge du NP. Et dès qu'elle est effectuée, le CN génère une nouvelle RT qui remplace la RT existante.

Comme la modification des accords inter-opérateurs ne modifie pas le vecteur d'une BE existante, l'identifiant de la RT reste donc inchangé et la RT reste trouvable. Elle sera donc mise à jour en gardant le même identifiant. Les différents identifiants du CN restent aussi inchangés puisqu'ils sont tous dérivés à partir de son *nodeId* primaire. Il en est de même pour les différents *nodeIds* d'un pair qui n'a pas quitté le système, tout comme des différents *objectIds* d'un même objet encore disponible.

Quand le CN a donc mis à jour sa RT, il va en informer les pairs de sa BE par un message de diffusion dans l'*overlay* de la BE. Ce message va

permettre aux pairs de se déconnecter de leurs anciennes IEs d'appartenance et de se joindre aux nouvelles. Mais dans un système dynamique, tous les pairs vont à un moment ou l'autre quitter le système avant de peut-être le rejoindre. Alors cette nouvelle adhésion se fait évidemment conformément aux mises à jour effectuées, puisque le nouveau pair en récupérant la RT y trouvera les informations mises à jour. Cette opération de remaniement (partiel) pourrait rendre certains *objets* rares momentanément indisponibles à certains niveaux secondaires.

Au cas où la modification des accords inter-NPs est due à l'émergence d'une nouvelle BE, cette émergence se manifestera par l'insertion dans le système d'un nouveau CN. Celui-ci initiera le sous-système hiérarchique correspondant et génèrera une RT correspondante qu'il mettra à disposition dans l'*overlay* global. Par ailleurs, comme chaque nœud se trouve dans une seule BE à la fois, alors si les nœuds de la nouvelle BE sont initialement dans une autre BE existante, ils ne le seront plus à présent et auront donc forcément déjà quittés le système P2P. Et s'ils veulent le rejoindre, ils le feront à partir de leur nouvelle BE d'appartenance (et ils vont donc chercher la nouvelle RT correspondante).

En garantissant la récupération d'un objet de l'entité où il a été trouvé, NETPOPPS offre un cloisonnement des ressources efficace et conforme aux intérêts d'un opérateur avec une gestion optimale des différents identifiants. Le cloisonnement réduit le temps de latence des messages et améliore le temps de récupération des données. Ces performances sont plus remarquables avec la possibilité qu'a un objet de garder un identifiant unique dans tout le système. De plus, la mise en partage dans l'*overlay* d'une entité locale d'une ressource récupérée à un autre niveau (intermédiaire ou global) plus large favorise la popularisation des objets dans le système.

Enfin, NETPOPPS offre à un opérateur la possibilité d'opérer un contrôle d'accès sur les pairs utilisant ses réseaux ou de leur offrir ses services propriétaires (qui peuvent être par exemple des services privilégiés et/ou payant). Ceci est techniquement réalisable grâce à la structure

minimale de la RT dans le cas d'un système actif. Outre les vecteurs des différentes entités d'appartenance, cette structure ne fournit que le *nodeId* primaire du CN.

#### CONCLUSION DU CHAPITRE –

*Ce chapitre a défini, spécifié et analysé NETPOPPS (NETwork Provider Oriented P2P System), un système P2P utilisant des DHTs et orienté opérateur de réseau. Ce système permet une coopération entre, d'une part, les flux P2P et, d'autre part, la topologie sous-jacente et les politiques de l'opérateur. NETPOPPS veille aussi à simplifier la gestion des différents identifiants d'un même élément (pair ou objet), en créant entre eux une relation mathématique simple, grâce à la technique de dérivation de clés, qui a été présentée au début du chapitre.*

*Le chapitre suivant établit une comparaison analytique et applicative de NETPOPPS avec CAP.*