

---

# Techniques d'amélioration des méthodes de gestion automatique des routeurs filtrants

Olivier Paul<sup>†</sup> — Maryline Laurent

ENST de Bretagne, Département RSM  
2 rue de la châtaigneraie, BP 78 - 35512 Cesson-sévigné, France  
(paul, mlaurent)@rennes.enst-bretagne.fr

---

*RÉSUMÉ.* Dans cet article nous décrivons une architecture assurant la configurations de routeurs filtrants dans un environnement internet. Il est bien connu que les performances des mécanismes de classification de paquets dépendent du nombre de règles utilisées pour définir une politique de contrôle d'accès. Par conséquent, une architecture de contrôle d'accès efficace se base sur l'utilisation de techniques de configuration efficaces du contrôle d'accès. Les approches actuelles pour résoudre ce problème se basent sur un modèle statique du réseau qui oblige l'officier de sécurité à faire un choix entre efficacité des mécanismes de contrôle d'accès et facilité de gestion. Notre architecture ne se base pas sur un modèle du réseau et ne souffre pas de cette limitation. Elle fournit trois critères d'optimisation assurant une configuration efficace des équipements. Des simulations montrent que notre architecture permet d'atteindre des complexités de configuration proches de celles pouvant être réalisées à la main par un officier de sécurité expérimenté.

*ABSTRACT.* The development of complex access control architectures raises the problem of their management. In this article, we describe an architecture providing packet filters configuration in Internet based networks. It is well known that the performance of the access control processing heavily depends on the number of rules used to define the access control service. Therefore an efficient access control architecture relies on a clever access control rule configuration. The current approaches to this problem are based on a static description of the network which can force security officers to choose between efficiency and manageability. Our distribution architecture doesn't rely on a model of the network and thus eliminates this limitation while proposing three optimisations in order to provide the access control processes with optimal configurations. Simulations show that our architecture succeeds in realising efficient configurations. The complexity of these configurations is close to the complexity found in configurations created by hand.

*MOTS-CLÉS :* Contrôle d'accès, Gestion, Sécurité.

*KEY WORDS:* Access Control, Management, Security.

---

<sup>†</sup> Ce travail est financé par une bourse DRET.

## 1. Introduction

Avec le développement d'Internet, la sécurité en général et le contrôle d'accès en particulier deviennent des problèmes essentiels dans le développement des réseaux de communication. C'est la raison pour laquelle de nombreux travaux ont été réalisés afin de produire des outils permettant de fournir le service de contrôle d'accès pour la plupart des types de réseaux. Cependant les progrès dans le domaine des réseaux ont également soulevé un autre sujet important : les performances des mécanismes de contrôle d'accès. Dans un passé récent, de nombreux travaux ont porté sur le développement d'algorithmes afin d'améliorer les performances des routeurs filtrants (La98), [Sri99]. Comme il est indiqué dans [Xu99], le problème de la classification de paquets sur  $d$  champs de données peut être considéré comme un problème classique de géométrie appelé problème de localisation d'un point. Ce problème consiste à trouver l'objet auquel appartient un point parmi  $N$  objets de dimension  $d$ . La forme générale de ce problème ne possède pas de solution satisfaisante sur le plan algorithmique lorsque  $d > 3$ . D'un côté, le meilleur algorithme en terme de complexité temporelle a une complexité de  $O(\log(N))$  mais a une complexité spatiale de  $O(N^d)$ . De l'autre le meilleur algorithme en terme de complexité spatiale a une complexité de  $O(N)$  mais possède une complexité temporelle de  $O(\log^{d-1} N)$ .

Même lorsque des compromis sont utilisés entre ces deux algorithmes, le nombre de règles ( $N$ ) qui est utilisé pour définir une politique de contrôle d'accès a un impact important soit sur la complexité temporelle du système, soit sur sa complexité spatiale. Une solution pour résoudre ce problème consiste à réduire le nombre de règles utilisées pour configurer à la main les équipements au moyen d'une configuration efficace. Cette méthode est utilisable pour un faible nombre d'équipements de contrôle d'accès mais devient inutilisable quand le nombre d'équipements de contrôle d'accès augmente.

Plusieurs solutions ont été proposées afin de réaliser la configuration d'équipements de contrôle d'accès de manière automatique ([Gu97], [Ft98], [Hyl98], [Ba99], [Hin99], [Plg99]). Le processus de distribution automatique est toujours basé sur la définition d'une interface générique qui permet de représenter au moyen d'un seul langage toutes les possibilités de filtrage proposées par des équipements possédant des interfaces diverses. La définition d'une politique de contrôle d'accès au moyen de ce type de langage se fait au moyen d'un ensemble de règles. Cependant les solutions peuvent être classées en fonction de leur processus de distribution en deux classes principales :

La première ([Ba99], [Hin99], [Plg99]) se base sur une description statique du réseau sous forme d'arbre. Elle permet des optimisations basées sur la topologie du réseau. Cependant cette solution oblige l'officier de sécurité à reconfigurer les équipements de sécurité après chaque changement de topologie.

La seconde ([Gu97], [Hyl98]) se base sur un mécanisme de distribution moins optimisé qui peut déboucher sur des configurations d'outils de contrôle d'accès très inefficaces. Cependant cette approche évite les problèmes causés par les changements de topologie.

Dans cet article, nous nous concentrons sur l'optimisation de la configuration du contrôle d'accès. Nous décrivons une architecture assurant la configuration automatique de routeurs filtrants dans un environnement internet. Le but de notre

architecture est d'assurer la configuration de chaque routeur filtrant de manière automatique avec un nombre minimal de règles et sans faire appel à une description statique du réseau.

La section 2 donne des détails sur les solutions actuellement proposées pour la configuration automatique d'équipements de contrôle d'accès. Nous décrivons ensuite une architecture de gestion du contrôle d'accès qui est basée sur des agents à placer dans les routeurs filtrants. La section 3 décrit cette architecture et explique comment distribuer une politique de contrôle d'accès. La section 4 présente les résultats de simulations de notre architecture au moyen du simulateur ns.

Pour conclure, la section 5 compare les avantages et inconvénients de cette architecture par rapport à l'existant et montre de quelle manière elle pourrait être améliorée.

## **2. Etat de l'art**

### **2.1. Solutions traditionnelles**

Le problème de la gestion des routeurs filtrants est un sous problème du problème de gestion des outils de contrôle d'accès. Deux approches sont actuellement utilisées pour la gestion de mécanismes de ce type. Dans ces approches, l'officier de sécurité définit une politique de contrôle d'accès pour chacun des modules et configure ensuite les modules. Cette configuration peut se faire :

- De manière directe, en se connectant à chacun des équipements et en configurant chaque module à la main. Cette méthode est intéressante quand un seul équipement contenant tous les modules doit être configuré mais devient inutilisable lorsque le nombre d'équipements augmente.
- De manière indirecte en utilisant une plate-forme d'administration du contrôle d'accès (Ma98). Cet outil permet de réaliser l'administration de plusieurs modules de contrôle d'accès à partir d'un seul équipement offrant généralement une interface générique à l'officier de sécurité. Cependant la décision de distribuer une partie de la politique de contrôle d'accès sur un module est prise par l'officier de sécurité et non par la plate-forme d'administration. De plus la genericité de l'interface fournie est généralement limitée par le fait qu'elle s'adresse à un ensemble de modules produits par un même constructeur et est donc de ce fait propriétaire.

Dans les deux cas l'officier de sécurité est responsable de la distribution de la politique de contrôle d'accès.

Le second problème posé par la gestion d'architectures distribuées est l'hétérogénéité des langages de configuration. Celle-ci est combinée au fait que le terme "firewall" est une expression générique qui désigne des mécanismes divers fournissant plusieurs types de services de contrôle d'accès. La configuration de ces mécanismes se fait généralement au moyen d'interfaces propriétaires. La figure 1 et la figure 2 montrent comment une simple règle de contrôle d'accès peut être exprimée au moyen de commandes relativement différentes en fonction de

l'équipement auquel elle est destinée (Un routeur Cisco et une station Linux dans notre cas).

Dans le cas du routeur on commence par autoriser les demandes de connexion TCP provenant de la station d'adresse 192.165.203.5 avec un port source supérieur à 1023 vers le port WWW (port 80) de n'importe quelle station. On autorise ensuite les paquets de requêtes correspondant à la connexion établie. On autorise enfin dans le sens du retour les paquets correspondant à des réponses. Il faut noter que les paquets de demande de connexion dans la direction serveur vers client sont naturellement bloqués. Cette propriété est due au fait que la politique de contrôle d'accès des routeurs Cisco est de type "Tout ce qui n'est pas explicitement autorisé est interdit."

Dans le cas de la station linux il n'est pas possible de désigner les paquets correspondant à une connexion établie. On est donc amené à interdire les paquets de demande de connexion dans le sens du retour (ligne 2) de manière explicite.

```
access-list 101 permit tcp 192.165.203.5 0.0.0.0 gt 1023
any eq 80
access-list 101 permit tcp 192.165.203.5 0.0.0.0 gt 1023
any eq 80 established
access-list 102 permit tcp any eq 80 192.165.203.5
0.0.0.0 gt 1023 established
```

**Figure 1.** Règles de contrôle d'accès pour un routeur Cisco.

Ces différences rendent la tâche d'administration de l'officier de sécurité plus difficile car elles peuvent provoquer l'introduction d'erreurs dans les fichiers de configuration qui peuvent être exploitées par des attaquants. De plus l'hétérogénéité oblige l'officier de sécurité à passer un temps important à apprendre ces langages. Ce temps pourrait être utilisé de manière plus utile d'un point de vue de la sécurité.

De plus ce type d'interface empêche l'officier de sécurité d'avoir une vision globale de la politique de sécurité qui sera appliquée au réseau. Ce que l'officier de sécurité configure en fait est un ensemble de petites politiques de contrôle d'accès appliquées en différents points du réseau. Ceci peut rendre sa tâche très difficile du fait des dépendances pouvant exister entre les politiques.

```
ipfwadm -F -a accept -b -P tcp -S 192.165.203.5
1024:65535 -D 0.0.0.0/0 80
ipfwadm -F -a deny -b -P tcp -S 0.0.0.0/0 80 -k -D
192.165.203.5 1024:65535
ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 80 -D
192.165.203.5 1024:65535
```

**Figure 2.** Règles de contrôle d'accès pour une station Linux utilisant ipfwadm.

Ces trois aspects montrent qu'une configuration efficace d'outils de contrôle d'accès distribué devient de plus en plus difficile. Il est donc nécessaire de rechercher de nouvelles méthodes rendant cette gestion aussi simple que possible.

## **2.2. Autres propositions**

Les problèmes présentés dans le paragraphe précédent ont donné lieu à un certain nombre de travaux destinés à faciliter la gestion de la sécurité en général et le contrôle d'accès en particulier dans le cadre de mécanismes répartis. Les solutions actuellement proposées peuvent être classifiées en quatre catégories en fonction de l'endroit où la distribution est décidée et des optimisations apportées.

### **Centralisé et manuel**

[Sam95] qui résume les résultats du projet européen SAMSON (Security and Management Services in Open Networks) décrit une architecture de gestion de la sécurité. Cette architecture développe plusieurs idées intéressantes. La première est d'offrir à l'officier de sécurité une interface générique pour la gestion des différents services de sécurité (authentification, contrôle d'accès sur les équipements, audit, gestion des clefs) et des différents mécanismes afin de résoudre le problème d'hétérogénéité. Un autre aspect intéressant est la possibilité d'utiliser plusieurs protocoles de gestion (CMIP et SNMP) de manière simultanée. Cependant le processus de configuration des équipements reste manuel puisque chaque module doit être configuré par l'officier de sécurité au travers de l'interface générique.

### **Centralisé et automatique avec optimisation faible**

[Gu97] est le premier article à introduire le concept de distribution automatisée. Ce processus de distribution est basé sur deux paramètres qui sont la politique de contrôle d'accès et un modèle du réseau. Cependant le modèle du réseau autorise les boucles à l'intérieur du réseau. Ces chemins cycliques génèrent des configurations non optimales car les règles de contrôle d'accès peuvent être attribuées à des équipements de contrôle d'accès qui ne sont pas topologiquement sur le chemin emprunté par une communication.

### **Centralisé et automatique avec optimisation forte**

Afin de résoudre le problème de l'inefficacité d'un modèle de réseau non acyclique, [Ba99], [Hin99] et [Plg99] se basent sur une description en arbre du réseau. Ils utilisent également une description des capacités de contrôle d'accès des équipements afin de garantir que les règles seront bien attribuées à des équipements qui pourront les appliquer. Cependant, cette approche oblige l'officier de sécurité à reconfigurer le modèle du réseau à chaque modification de la topologie afin de refléter l'état du réseau. Il faut également noter qu'un modèle périmé peut entraîner des trous de sécurité importants puisque les communications peuvent passer par des équipements non configurés. Malgré cet inconvénient, cette approche est la plus utilisée aujourd'hui.

### **Distribué et automatique sans optimisation**

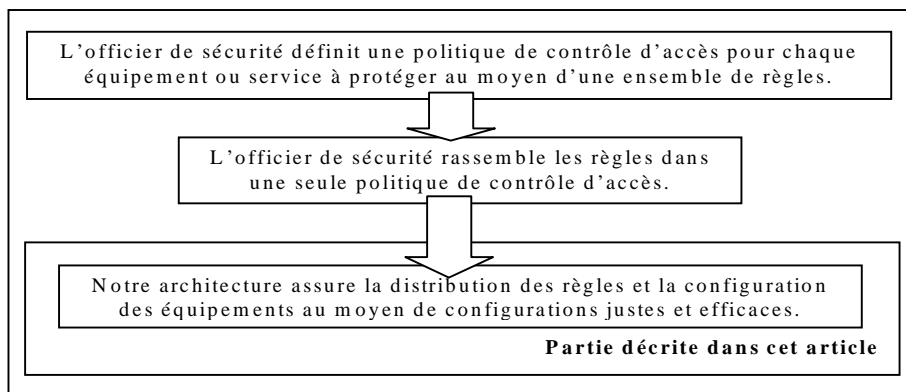
[Hyl98] reprend l'idée d'une interface générique et suggère que les règles de contrôle d'accès soient diffusées dans le réseau à la manière des informations de routage. Les auteurs présentent comme exemple un protocole de distribution des règles de contrôle d'accès pour les routeurs filtrants (PFIP, Packet Filter

Information Protocol). Cette approche est intéressante car elle présente une méthode distribuée de configuration d'équipements eux-mêmes distribués. Le protocole PFIP permet aux équipements de contrôle d'accès d'adapter les règles de contrôle d'accès qu'ils appliquent en fonction de la topologie du réseau. Les auteurs donnent également des informations sur une implémentation éventuelle du protocole. Cependant la méthode d'optimisation de la configuration des équipements vise à optimiser l'utilisation du réseau et non le processus de contrôle d'accès lui-même.

Pour résumer ce paragraphe, nous pouvons constater que la majorité des propositions se concentrent sur la définition d'un langage générique de contrôle d'accès et fournissent moins d'information sur l'optimisation du processus de distribution. Nous pensons que plusieurs améliorations sont possibles dans ce domaine. C'est la raison pour laquelle nous abandonnons le problème de la définition d'un langage générique de description du contrôle d'accès pour nous concentrer sur le problème de l'optimisation du processus de distribution. Nous montrons comment améliorer et implémenter les optimisations actuelles sans nous baser sur une description du réseau. Notre approche permet aux routeurs filtrants d'adapter leur configuration à la topologie du réseau sans interaction avec l'officier de sécurité. De plus notre architecture n'est pas basée sur un équipement de gestion centralisé mais sur l'utilisation d'agents interagissant entre eux. En conséquence notre architecture est plus réactive et plus extensible.

Dans les sections suivantes, nous supposons qu'un langage générique de définition du contrôle d'accès a été défini. Ce langage permet à l'officier de sécurité de spécifier celle-ci au moyen d'un ensemble de règles.

### 3. Solution proposée



**Figure 3.** Processus de configuration du contrôle d'accès.

### 3.1. Architecture générale

Afin de clarifier le positionnement de notre processus de distribution, nous détaillons en figure 3 les différentes actions réalisées par un officier de sécurité utilisant notre architecture.

Comme nos optimisations se basent sur le contenu des règles de contrôle d'accès, nous devons spécifier comment ces règles peuvent être définies. Chaque règle est constituée d'un ensemble de conditions et d'une action. Chaque condition est faite d'un objet de contrôle d'accès, d'un opérateur et d'une valeur ou d'un ensemble de valeurs de contrôle d'accès. L'action spécifie généralement si la communication doit être autorisée ou interdite.

Notre architecture est basée sur l'utilisation d'agents se plaçant sur les équipements fournissant le service de contrôle d'accès. Ces agents interagissent avec les équipements sur lesquels ils sont placés afin de configurer les mécanismes de contrôle d'accès de la manière la plus performante possible.

L'objectif général de l'application des règles spécifiées par la politique de contrôle d'accès au niveau des équipements de contrôle d'accès est qu'un nombre minimal de règles de contrôle d'accès soit appliqué sur chaque équipement. Pour cela, nous définissons un ensemble d'assertions qui assure cette propriété lorsqu'elles sont appliquées :

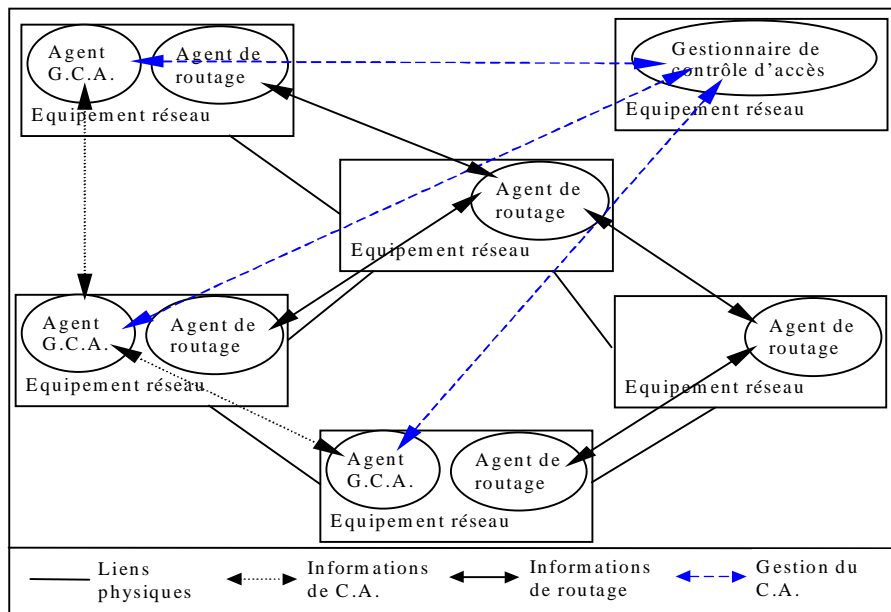
- (Assertion 1) : Une règle ne peut être attribuée qu'à des modules de contrôle d'accès contenant les objets de contrôle d'accès utilisés par la règle.
- (Assertion 2) : Une règle ne peut être attribuée à un module contenu dans un équipement que si celui-ci est situé sur le chemin entre la source et la destination décrites par la règle.

Une meilleure connaissance du type de politique de contrôle d'accès permet de donner des propriétés supplémentaires. Par exemple, dans le cas d'une politique de type «Tout ce qui n'est pas explicitement autorisé est interdit», qui est le type de politique le plus répandu, il est possible de ne spécifier les interdictions qu'en un point du chemin qui interconnecte la source et la destination relatifs à une règle. Ceci se justifie par le fait que dans ce type de politique, chaque règle d'interdiction décrit un sous-ensemble d'un ensemble décrit par une règle d'autorisation. De plus afin d'assurer une distribution maximale des règles et du fait de la structure généralement arborescente des réseaux, il est important que le placement des règles se fasse sur les équipements les plus proches des équipements source et destination décrits par chaque règle.

- (Assertion 3): Si une règle d'interdiction peut être attribuée à des modules situés sur plusieurs équipements en cascade, la règle ne doit être attribuée qu'au module situé sur l'équipement le plus proche des extrémités.

La traduction de ces assertions se fait par la prise en compte de plusieurs paramètres : La topologie du réseau qui varie en fonction des informations de routage, les capacités de l'équipement en terme de contrôle d'accès, la place de l'équipement dans la topologie du réseau et enfin la configuration des autres équipements de contrôle d'accès.

Ces paramètres impliquent des interactions entre les agents de configuration et d'autres éléments du réseau. Certaines de celles-ci, détaillées dans la section 3.2 sont internes à un équipement : entre l'agent de configuration et l'agent de routage correspondant ainsi qu'entre l'agent de configuration et les mécanismes de contrôle d'accès.



**Figure 4.** Relations entre éléments externes.

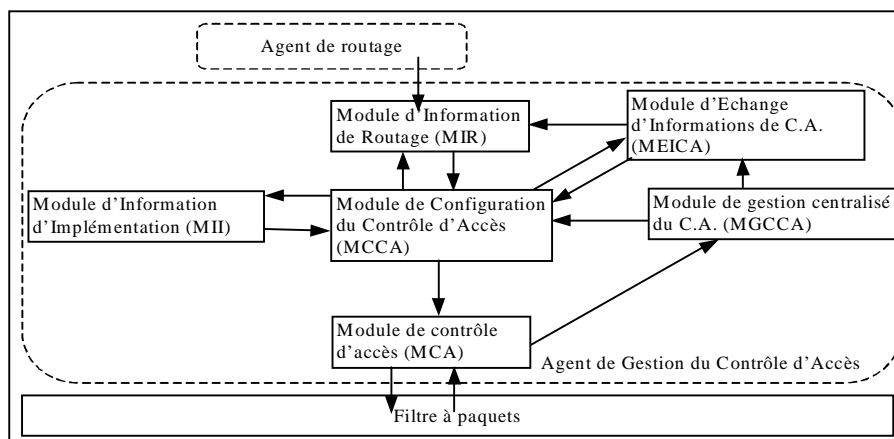
D'autres interactions, détaillées figure 4, sont externes :

- Entre les agents de gestion du contrôle d'accès (G.C.A.) afin d'assurer une configuration efficace.
- Entre les agents et le gestionnaire centralisé de contrôle d'accès. Celui-ci assure la distribution uniforme de toutes les règles de contrôle d'accès sur les agents et la récupération des résultats de l'application de celles-ci. Le protocole utilisé entre les agents et le gestionnaire doit assurer l'intégrité, l'authentification, la confidentialité et le contrôle d'accès sur les informations transportées. Les protocoles SNMPv2\* [Sta98] et SNMPv3 [Ba98] semblent être de bons candidats. Ils exigent cependant de gérer les informations de contrôle d'accès sous forme de bases de données de gestion (MIB) au niveau des agents. Ces informations comportent d'une part les règles de contrôle d'accès et d'autre part les résultats relatifs à l'application de ces règles.

Nous décrivons dans les sections suivantes comment ces règles peuvent être mises en œuvre au moyen de notre architecture.

### 3.2. Architecture fonctionnelle

Comme on peut le voir figure 5, notre architecture se base sur l'utilisation de modules interagissant entre eux.



**Figure 5.** Architecture fonctionnelle d'un agent de contrôle d'accès.

#### Le module de contrôle d'accès (MCA)

Ce module représente les mécanismes de contrôle d'accès implémentés par l'équipement auprès de l'agent de contrôle d'accès. Ce module reçoit les règles de contrôle d'accès de la part du module de configuration du contrôle d'accès (MCCA). Ces règles sont traduites par le MCA dans les syntaxes réelles des commandes nécessaires à la configuration des mécanismes implémentés. En retour de ces commandes le MCA reçoit des résultats correspondant à l'application des commandes et transmet ces résultats au module de gestion centralisée du contrôle d'accès (MGCCA).

#### Le module d'information d'implémentation (MII)

Le MII décrit au MCCA les capacités de l'équipement sur lequel se situe l'agent en terme de contrôle d'accès. Pour cela il possède une table de correspondance entre les paramètres présents dans les règles de contrôle d'accès et les mécanismes implémentés au niveau du système. La configuration de cette table se fait par l'officier de sécurité de manière manuelle ou de manière automatique par le logiciel d'installation des outils de contrôle d'accès. Lors de modifications dans cette base de donnée, le MCCA est avertit par le MII.

#### Le module de gestion centralisée du contrôle d'accès (MGCCA)

Ce module a pour objectif de faire le lien entre l'agent et le gestionnaire centralisé de contrôle d'accès. Pour cela il gère une base de donnée de gestion (MIB) qui est mise à jour par le gestionnaire. Celui-ci distribue au MGCCA toutes les règles de la politique de sécurité. Le MGCCA avertit le MCCA de ces mises à jour en lui fournissant toutes les règles modifiées. En retour de celles-ci, le

MGCCA reçoit les résultats correspondants de la part du MCA. Ces résultats sont stockés de manière incrémentale dans une table de la MIB et sont récupérés par le gestionnaire de manière périodique.

#### **Le module d'échange d'informations sur le contrôle d'accès (MEICA)**

Le MEICA a pour objectif de répondre à la question suivante du MCCA : «Existe-t'il un équipement mieux positionné qui applique déjà la règle  $r$  ?». Afin de répondre à cette question, le MEICA interagit avec les MEICAs des agents voisins au moyen du protocole d'échange d'informations de contrôle d'accès. Ce protocole permet l'élection d'un agent choisi pour appliquer la règle de contrôle d'accès. Pour cela, le MEICA envoie à chaque changement de configuration les informations suivantes à ses voisins : Identificateur de la règle, Adresse de l'agent appliquant la règle, Distance. Pour chaque règle  $r$ , le MEICA compare les informations fournies par ses voisins avec les siennes et en déduit si l'agent qu'il représente se trouve en position optimale pour appliquer la règle. Une position optimale signifie que la valeur minimale des distances entre la source ou la destination relative à la règle et notre MEICA est la plus faible. Cette distance est donnée par le MIR. Dans le cas de changements de topologie ou dans la configuration des équipements de contrôle d'accès provoquant une modification des informations de contrôle d'accès, le MEICA avertit le MCCA afin que celui-ci revoit sa configuration.

#### **Le module d'informations de routage (MIR)**

Le module MIR a un fonctionnement qui peut varier énormément en fonction du protocole de routage qui lui est associé. Cependant les fonctions qu'il remplit sont toujours les mêmes. D'une part, il est utilisé par le MCCA afin de savoir si la communication relative à une règle  $r$  passe par l'équipement sur lequel est installé l'agent. D'autre part le MIR est chargé par le MEICA de calculer la distance entre l'équipement supportant l'agent et un équipement  $y$ . Enfin lors de modifications topologiques, le MIR avertit le MCCA afin que celui-ci prenne en compte celles-ci.

#### **Le module de configuration du contrôle d'accès (MCCA)**

Le module MCCA est le module central de notre architecture. Il est averti de l'arrivée de chaque nouvelle règle de contrôle d'accès par le MGCCA. Pour chacune de ces règles, il va appliquer les trois assertions définies dans la section précédente. Pour cela, il interagit avec les modules MII, MIR, MEICA. L'algorithme de fonctionnement général de ce module est le suivant. Pour chaque règle reçue, le MCCA vérifie si celle-ci peut être appliquée à l'équipement au moyen du MII (assertion 1). Il vérifie ensuite au moyen du MIR que l'équipement peut se situer sur une des routes décrites par la règle (assertion 2). Si la règle de contrôle d'accès est une règle d'interdiction il s'adresse alors au MEICA afin de savoir si un équipement plus proche de la source ou de la destination décrite par la règle l'applique déjà (assertion 3). Les règles n'exprimant pas de notion de source et de destination sont appliquées sans cette vérification. Si toutes ces conditions sont vérifiées, le MCCA passe alors la règle au MCA afin que celui-ci l'implémente.

Les simulations fournies à la section 4 montrent que pour un ensemble de règles et une topologie donnés, la configuration de chaque équipement converge vers un état stable qui satisfait nos trois assertions.

## 4. Simulations

### 4.1. Implémentation

Le simulateur ns [Ns99] est un simulateur à événement discret dédié à la recherche en réseaux. L'implémentation de ns est faite de deux parties. Une première partie en O-Tcl (Object Tcl) implémente des fonctions simples et permet à l'utilisateur de manipuler des objets représentant des équipements réseau de manière simple. La plupart de ces objets sont implémentés en C++ dans la seconde partie du simulateur pour des raisons d'efficacité.

Au moyen du simulateur ns, nous avons réalisé une implémentation de l'architecture présentée dans la section précédente. Le lecteur intéressé peut trouver un package complet contenant l'implémentation et plusieurs suites de test sur notre page web [Pa99]. Cette implémentation est relativement compacte. Elle contient deux parties. La première, composée d'une centaine de lignes de C++ est dédiée à la transmission des paquets produits par le PEICA. La seconde partie, composée d'un millier de lignes d'O-Tcl implémente les modules décrits dans la section précédente. La raison pour laquelle nous avons codé ces modules en O-Tcl est que nos modules doivent interagir avec les modules de routage et que ces modules sont implémentés dans la partie O-Tcl du simulateur ns.

Dans les sections suivantes, nous montrons comment des règles de contrôle d'accès peuvent être distribuées au moyen de l'architecture que nous avons développé. Nous définissons ensuite une politique « typique » de contrôle d'accès et examinons les résultats de sa distribution sur différents types de réseaux. Les résultats nous permettent de montrer l'efficacité de notre méthode de distribution et d'analyser les améliorations fournies par chaque assertion.

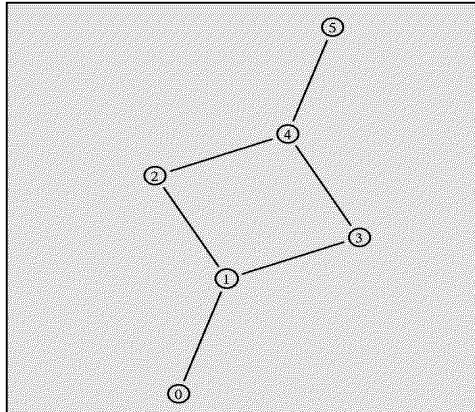
### 4.2. Exemples de distribution

Dans cette partie nous expliquons comment les règles de contrôle d'accès sont distribuées par notre architecture de gestion du contrôle d'accès et comment celle-ci réagit au changement de topologie.

La topologie utilisée pour ces exemples est très simple. Elle est constituée de deux nœuds terminaux interconnectés au moyen d'un petit réseau maillé. Cette topologie est présentée figure 6. Nous supposons que les nœuds 1 à 4 ont les mêmes capacités de contrôle d'accès. Celles-ci leur permettent de fournir n'importe quel service de contrôle d'accès. Les nœuds 0 et 5 sont des postes clients qui n'ont aucune capacité de contrôle d'accès.

Afin de tester les changements de topologie, nous configurons notre programme de simulation pour désactiver le lien entre les nœuds 1 et 2 après 1sec, nous désactivons ensuite le lien entre 1 et 3 et rétablissons le lien entre 1 et 2 après 1.5 sec. Cette opération provoque le changement de route entre les nœuds 0 et 5 en la faisant passer d'un côté du réseau à l'autre. Les règles sont envoyées à tous les

nœuds 0.5 sec après le début de la simulation. Nous fournissons trois exemples en fonction d'une classification qui sera expliquée dans la section suivante.



**Figure 6.** *Topologie simple.*

#### **Règle spécifiée d'autorisation**

Le but de la première règle (appelée règle 0) est de permettre les communications d'un client WWW situé sur le nœud 0 vers un serveur WWW situé sur le nœud 5. L'exécution de la simulation donne les résultats suivants :

```

Node 1 configuring rule 0.
Node 2 configuring rule 0.
Node 4 configuring rule 0.
Node 3 configuring rule 0.
Node 2 deleting rule 0.
Node 2 configuring rule 0.
Node 3 deleting rule 0.
  
```

La conséquence du changement de topologie est évidente, la configuration du contrôle d'accès suit le changement de topologie. Le placement des règles reste optimal puisque seuls les routeurs filtrants situés sur le chemin entre la source et la destination sont configurés.

#### **Règle spécifiée d'interdiction**

Le but de la première règle (appelée règle 1) est d'interdire les communications d'un client WWW situé sur le nœud 5 vers un serveur WWW situé sur le nœud 0. L'exécution de la simulation donne les résultats suivants :

```

Node 1 configuring rule 1.
  
```

Comme nous l'avons dit dans la section précédente, une seule règle d'interdiction suffit à bloquer la communication. C'est la raison pour laquelle seul le nœud 1 est configuré.

### Règle non spécifiée

Le but de la troisième règle (appelée règle 2) est d'interdire les messages ICMP redirect. L'exécution de la simulation donne les résultats suivants :

```
Node 1 configuring rule 2.
Node 2 configuring rule 2.
Node 3 configuring rule 2.
Node 4 configuring rule 2.
```

Les règles qui ne fournissent pas d'adresses source et destination doivent être attribuées à tous les nœuds fournissant le service de contrôle d'accès. Cependant un officier de sécurité peut limiter leur distribution en enlevant la capacité de contrôle d'accès correspondante sur les équipements adéquats.

### 4.3. Tests de performance

#### Modélisation de la politique de contrôle d'accès.

Afin d'obtenir des résultats de simulation satisfaisants, nous utilisons notre simulateur pour représenter une politique réelle de contrôle d'accès. Comme les politiques de contrôle d'accès peuvent varier fortement d'un site à un autre, il n'est pas évident de définir une politique « typique » de contrôle d'accès. De plus comme nous l'avons noté en section 2, le langage utilisé pour définir une politique de contrôle d'accès peut apporter des modifications importantes à la façon dont la politique sera exprimée. Nous avons donc pris des exemples de politiques provenant de [Ches94] et [Cha95] qui présentent des politiques « typiques » de contrôle d'accès pour divers services utilisés dans internet et nous avons ajouté certaines modifications afin de refléter ce que l'on peut trouver dans une configuration réelle. La politique obtenue contient 80 règles. 10 de ces règles sont destinées à prévenir des attaques qui ne sont pas liées à des adresses spécifiques. Ce type de règle peut être par exemple utilisé pour rejeter les paquets IP utilisant le source-routing ou les messages ICMP redirect. 70 règles sont dédiées au contrôle d'accès aux services internet (mail, dns, www, news, time, ftp, telnet, ...). Ces règles sont toujours liées à des adresses ou à des espaces d'adresse.

Afin de modéliser ces règles, nous classons les règles de contrôle d'accès suivant deux critères. Le premier est l'action associée à la règle. Celle-ci peut être une interdiction ou une autorisation de la communication. En conséquence nous avons deux classes de règles appelées ALLOW et DENY. Le second est les informations d'adressage associées à la règle. Les règles qui fournissent des informations d'adressage sont appelées « spécifiées » alors que les règles qui n'en fournissent pas sont appelées « non spécifiées ».

Type de règle	ALLOW	DENY
Non spécifiée	5	5
Spécifiée	60	10

Tableau 1. Classement des règles.

### Exemple de distribution dans un réseau maillé

Notre exemple est basé sur un réseau maillé. Celui-ci pourrait représenter le réseau privé d'une entreprise. Il est constitué de 39 nœuds. 27 de ces nœuds sont terminaux et représentent des sous réseaux. Ces sous réseaux sont constitués de réseaux locaux interconnectés par un WAN. La topologie globale est décrite en figure 7. Les 12 nœuds non terminaux représentent le cœur du réseau. Ces deux classes de nœuds ont des requêtes différentes en matière de contrôle d'accès. Les nœuds terminaux sont supposés fournir un service de contrôle d'accès pour tous les équipements qu'ils représentent alors que les nœuds non terminaux doivent uniquement être protégés contre des attaques de type « dénis de service ». En conséquence, les nœuds non terminaux implémentent uniquement les règles non spécifiées alors que les nœuds terminaux implémentent toute la politique.

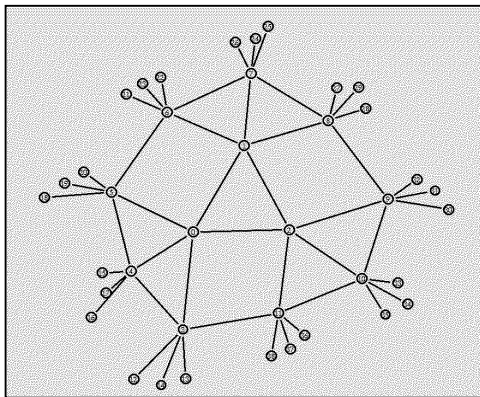


Figure 7. Topologie du réseau

L'instanciation de notre politique de contrôle d'accès est présentée dans le tableau 2. Comme les règles spécifiées sont dédiées à un réseau à protéger, la politique est composée de 27 ensembles de 70 règles de contrôle d'accès. Les règles non spécifiées sont par contre les mêmes pour tous les routeurs filtrants et sont donc représentées une seule fois. La politique obtenue est composée de 1900 règles.

Type de règle	ALLOW	DENY	Total
Non spécifiée	5	5	10
Spécifié	60	10	1890
<b>Total</b>	1625	275	1900

Tableau 2. Classement de l'instanciation des règles.

Les résultats de nos simulations sont présentés dans le tableau 3. Ces résultats montrent l'efficacité relative de l'utilisation de chaque assertion. L'optimisation la plus intéressante est celle basée sur les informations de routage. Les capacités des nœuds apportent également un gain intéressant. L'optimisation basée sur le type d'action apporte des améliorations plus faibles. Cependant une simulation avec une politique comprenant plus de règles « DENY » aurait donné des meilleurs résultats.

Méthode de distribution	Nombre de règles	Nombre de règles/ nœud	Diminution (%)
Brute	74100	1900	0
Basée sur les capacités	51420	1318	30.6
Basée sur les informations de routage	9691	248	86.9
Basée sur les infos de routage et l'action	8679	222	88.3
Optimisation complète	3842	99	94.8

**Tableau 3.** *Efficacité des différents critères.*

Les améliorations apportées par nos techniques d'optimisation sont importantes puisque nous parvenons à diviser par 20 le nombre de règles. D'une manière plus générale la complexité des politiques implémentées par chaque routeur filtrant est réduite de  $O(n \cdot m)$  avec une configuration brute à  $O(n)$  (où  $n$  est le nombre de règle pour protéger un nœud terminal et  $m$  le nombre de réseaux à protéger). Cependant nos simulations montrent que nos optimisations ne parviennent pas à configurer chaque nœud terminal avec  $n$  règles. Ceci s'explique par le fait que chaque règle spécifiée de type « ALLOW » génère la configuration de deux filtres de paquets. Le premier étant celui le plus près de la source et le second celui le plus près de la destination. Ce problème pourrait être réglé par l'agrégation de règles au moment de la définition de la politique de sécurité globale.

## 5. Conclusion

Dans cet article, nous avons présenté une architecture définie pour distribuer et configurer automatiquement une politique de contrôle d'accès dans un réseau comprenant plusieurs routeurs filtrants. Nous montrons au travers de simulations que notre architecture est capable de configurer ces équipements efficacement. En comparaison avec les approches existantes notre architecture présente plusieurs avantages :

- Elle permet à l'officier de sécurité de définir une politique de contrôle d'accès et de distribuer celle-ci de manière automatique sur les routeurs filtrants.
- Les configurations produites par notre architecture sont d'une efficacité comparable à celle produite par un administrateur expérimenté.
- L'officier de sécurité n'a pas à gérer la complexité introduite par la topologie du réseau car la distribution des règles s'adapte automatiquement à celle-ci.
- La répartition du processus de distribution le rend plus réactif et extensible.
- L'officier de sécurité peut contrôler l'application des règles au moyen des résultats renvoyés par les agents.

Notre travail pourrait être poursuivi dans plusieurs directions. La première est d'améliorer la sécurité de notre architecture en étendant le protocole PEICA afin qu'il fournisse les services d'authentification et d'intégrité. La seconde voie serait de s'assurer de la validité de notre architecture au moyen de preuves par modèles. Il serait enfin intéressant de tester notre architecture avec des politiques réelles de contrôle d'accès pour en mesurer l'efficacité.

## 6. Bibliographie

- [Ba98] : Basking in Glory-SNMPv3, Dan Backman, Network Computing, Août 1998.
- [Ba99] : Firmato: A Novel Firewall Management Toolkit, Y. Bartal, A. Mayer, K. Nissim, A. Wool, 20th IEEE Symposium on Security and Privacy S&P99, Oakland, Mai 1999.
- [Cha95] : Building Internet Firewalls, B. Chapman, E. Zwicky, O'Reilly, 1995.
- [Ches94] : Firewalls and internet security, repelling the wily hacker, B. Cheswick, S. Bellonin, Addison-Wesley publishing company, 1994.
- [Ft98] : Integrated Management of Network and Host Based Security Mechanisms, R. Falk, M. Trommer, 3rd Australasian Conference on Information Security and Privacy ACISP'98, Brisbane, Australia, 13.-15. Juillet 1998.
- [Gu97] : Filtering Postures: Local Enforcement for Global Policies, Joshua D. Guttman, IEEE Symposium on Security and Privacy, Oakland, CA, Mai 1997.
- [Hin99] : Policy-Based Management: Bridging the Gap, Susan Hinrichs, 15th Annual Computer Security Applications Conference, Phoenix, Décembre 1999.
- [Hyl98] : Management of Network Security Application, P. Hyland, R. Sandhu, 21st National Information Systems Security Conference, Octobre 1998.
- [La98] : High-Speed Policy based Packet Forwarding Using Efficient Multi-dimensional Range Matching, T.V. Lakshman, D. Stiliadis, ACM Sigcomm'98, Septembre 1998.
- [Ma98] : M-wall firewall administrator documentation, Matranet, 1998.
- [Ns99] : ns Notes and Documents, Kevin Fall, Kannan Varadhan, Septembre 1999.
- [Pa99] : [www.rennes.enst-bretagne.fr/~paul/acm.zip](http://www.rennes.enst-bretagne.fr/~paul/acm.zip).
- [Plg99] : An Asynchronous and Distributed Access Control Architecture for IP over ATM networks, Olivier Paul, Maryline Laurent, Sylvain Gombault, 15th Annual Computer Security Applications Conference, Phoenix, Décembre 1999.
- [Sam95] : Samson, Security and Management Services in Open Networks, Final Report, Michael Steinacker, RACE R2058 Project, Janvier 1995.
- [Sri99] : Packet Classification using Tuple Space Search, V. Srinivasan, S. Suri, G. Varghese, ACM Sigcomm'99, Septembre 1999.
- [Sta93] : SNMP, SNMPv2 and CMIP, The practical guide to network management Standards. William Stallings. Addison-Wesley. 1993.
- [Xu99] : A Novel Hardware Cache Architecture to support layer-four Packet Classification at Memory Access Speeds, J. Xu, M. Singhal, J. Degroat, Technical report, The Ohio State University, Février 1999.