



**HAL**  
open science

## **MINNIE : enfin un monde SDN sans (trop de) règles**

Myriana Rifai, Nicolas Huin, Christelle Caillouet, Frédéric Giroire, Joanna Moulhierac, Dino Lopez Pacheco, Guillaume Urvoy-Keller

### ► To cite this version:

Myriana Rifai, Nicolas Huin, Christelle Caillouet, Frédéric Giroire, Joanna Moulhierac, et al.. MINNIE : enfin un monde SDN sans (trop de) règles. ALGOTEL 2016 - 18èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2016, Bayonne, France. hal-01304687

**HAL Id: hal-01304687**

**<https://inria.hal.science/hal-01304687>**

Submitted on 20 Apr 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# *MINNIE : enfin un monde SDN sans (trop de) règles*

Myriana Rifai<sup>1</sup>, Nicolas Huin<sup>1,2</sup>, Christelle Caillouet<sup>1,2</sup>, Frederic Giroire<sup>1,2</sup>, Joanna Moulhierac<sup>1,2</sup>, Dino Lopez Pacheco<sup>1</sup> et Guillaume Urvoy-Keller<sup>1</sup>

<sup>1</sup> *University Nice Sophia Antipolis, Laboratoire I3S, UMR 7172, CNRS, 06900 Sophia Antipolis, France.*

<sup>2</sup> *INRIA, COATI*

---

Le paradigme des Software Defined Networks (SDNs), ou réseaux programmables, gagne en popularité ces dernières années. Il permet d'obtenir un meilleur contrôle sur le réseau en regroupant l'intelligence des commutateurs ou des routeurs sur un ou plusieurs contrôleurs. Cependant, ce gain se fait au détriment de la taille disponible pour les tables de "forwarding" des équipements SDN qui utilisent des mémoires spécifiques de petite taille.

Dans ce papier, nous proposons MINNIE, un algorithme de routage SDN se fondant sur des techniques de compression de règles pour réduire la taille des tables. Nous le validons par simulation pour différentes topologies de réseaux de centres de données, et par expérimentation sur une plateforme de type fat tree composée de 20 commutateurs. Côté simulation, nous montrons que MINNIE peut supporter aux alentours d'un million de flots lorsque la limite est de seulement 1000 règles par table, et avec des temps de calcul (routage et compression) négligeables. Côté expérimentation, nous montrons que, sans MINNIE, la limite de règles peut être rapidement atteinte avec un faible nombre de clients, ce qui accroît le délai sur le réseau. Avec MINNIE, le nombre de règles est réduit de manière importante sans introduire de perte de paquets ni de délai supplémentaire visible. Dans les deux cas, MINNIE affiche des taux de compression de tables entre 70 et 99%.

---

## 1 Introduction

Depuis quelques années, le paradigme SDN attire l'attention des chercheurs et des industriels ; et de plus en plus d'équipements supportant ce type de réseau programmable voient le jour. Contrairement aux réseaux classiques où le routage est calculé grâce à des protocoles distribués, SDN se base sur une séparation du plan de contrôle et du plan de données : un ou plusieurs contrôleurs sont placés sur le réseau et décident de l'acheminement des paquets. Ils transmettent alors aux commutateurs (qui sont de simples équipements de transmission de paquets, sans intelligence) les règles de "forwarding".

Les règles de routage utilisées dans les réseaux SDN sont plus complexes que celles des réseaux classiques. Dans OpenFlow 1.0, la reconnaissance de paquet peut s'effectuer sur un maximum de 12 champs. La complexité de ces règles nécessite l'utilisation de Ternary Content Adressable Memory (TCAM) qui est malheureusement plus chère et plus gourmande que le type de mémoire utilisé sur les réseaux classiques. La taille des tables de routage pouvant être parcourues s'en retrouve alors grandement réduite (de l'ordre de 750 à 4000 règles [nec]). Cette limitation est une contrainte importante pour le déploiement des SDNs.

Afin d'assurer un routage efficace dans les réseaux SDN, les auteurs de [CLENR14] proposent de répartir les chemins sur l'ensemble du réseau, en prenant en compte l'utilisation des tables de routage dans la répartition. Cela introduit un compromis entre Qualité de Service (QoS) et contrainte de taille qui pourrait, dans certains cas, dégrader la QoS lorsque l'utilisation des tables est basse. D'autres travaux tels que [MLT12, BK14] utilisent des modifications d'en-tête sur les paquets afin de réduire la taille nécessaire à la représentation d'un flot. Ces modifications nécessitent des changements au niveau de l'ajout des règles dans les tables ainsi que dans le traitement des paquets, ce qui ralentit grandement la vitesse de traitement.

Dans ce papier, nous présentons MINNIE, une solution de routage SDN qui utilise la compression de règles consistant à utiliser des règles d'agrégation pour réduire leur nombre dans les tables. L'agrégation peut s'effectuer par source (i.e.  $(s, *, p)$ ), par destination (i.e.  $(*, t, p)$ ) ou grâce à la règle par défaut (i.e.

Topologie	# de serveurs	# de commutateurs	# de liens	Degré moyen	# de flots			# de regles avec comp.			Moyen Comp. Ratio	Temps de calcul en moyenne (ms)	
					Total	par commutateurs Max	Moyen	Total	Max	Moyen		Chemin	Comp.
Group 1													
4-Fat tree (64)	1024	20	1056	54.4	917504	454244	216268	8923	999	446	~ 99.60	0.17	13
8-Fat tree (8)	1024	80	1280	19.2	1015808	649044	61030	25853	999	323	~ 99.61	0.21	7
16-Fat tree (1)	1024	320	3072	16	1040384	630998	15897	97173	999	303	~ 98.42	0.30	5
VL2(16, 16, 14)	896	88	384	16	790272	261266	42906	59237	1000	673	~ 97.90	0.15	4
VL2(8, 8, 64)	1024	28	612	~ 41.1	983040	423752	161499	22394	1000	799	~ 99.45	0.19	11
VL2(16, 16, 16)	1024	88	1152	~ 17.5	1032192	276575	56040	57078	1000	648	~ 98.39	0.18	4
Group 2													
Dcell(32, 1)	1056	33	1584	~ 2.91	1114080	63787	4893	123655	1000	113	~ 97.23	0.09	2
Dcell(5, 2)	930	186	1860	~ 3.33	863970	11995	5716	717018	994	642	~ 87.84	0.19	2
BCube(32, 1)	1024	64	2048	~ 3.77	1047552	37738	3734	358204	999	329	~ 86.04	0.19	2
BCube(10, 2)	1000	300	3000	~ 4.62	999000	10683	4153	849316	998	653	~ 80.85	0.25	2
BCube(6, 3)	1296	864	5184	4.8	1678320	7852	5184	1795400	991	831	~ 83.18	0.49	4

TABLE 1: Résultats de MINNIE sur différentes topologies avec environ 1000 serveurs.

( $*, *, p$ ). Afin de distinguer les cas où plusieurs règles correspondent à un flot, les règles ont des priorités par rapport aux autres. Dans [GMP14], seule l'utilisation de la règle par défaut est considérée. Avec MINNIE, nous considérons une compression plus forte, multi-champs, où le champ source ou destination peut aussi être compressé. Notez que notre solution peut être aussi appliquée sur d'autres couples de champs tels que le ToS, le protocole de transport ou autre.

Puisque le problème de compression de table est NP-Complet, MINNIE implémente une heuristique de compression qui crée trois tables compressées différentes, utilisant seulement l'agrégation par source pour la première, par destination pour la deuxième, ou la règle par défaut pour la dernière, et choisit la plus petite des trois tables. Cette heuristique est une 3-approximation [GHM15]. Puis MINNIE utilise une heuristique de routage où la métrique des poids dépend de l'utilisation des liens et de la taille des tables. Pour une définition formelle de MINNIE, nous redirigeons le lecteur vers [RHC<sup>+</sup>16].

En Section 2, nous présentons les résultats de MINNIE obtenus par simulation sur plusieurs topologies de centres de données et montrons que MINNIE passe à l'échelle en supportant jusqu'à 1 million de flots avec une limite de 1000 règles par table. En Section 3, nous déployons notre solution sur une plateforme SDN de type fat tree et montrons qu'aucun délai ni perte de paquets n'est introduit par la compression des tables.

## 2 Simulations

Nous nous intéressons au comportement de MINNIE sur quatre des architectures les plus fréquentes des centres de données, voir [GHJ<sup>+</sup>09] : Fat tree, VL2, BCube et DCell. Ces topologies peuvent être séparées en 2 groupes. Dans le premier groupe, constitué de fat tree et de VL2, les serveurs agissent en tant que simples hôtes. Dans le second groupe (Dcell et BCube), les serveurs occupent aussi le rôle de commutateur.

Dans la Figure 1, nous comparons le nombre maximal de règles installées sur un commutateur dans une topologie de fat tree entre trois scénarios. Dans le premier scénario, aucune compression n'est effectuée, seul le module de routage est utilisé. Dans le deuxième scénario, le module de compression est appelé lorsque tous les chemins ont été installés par chaque table. Le troisième scénario correspond à l'utilisation complète de MINNIE. Avec une limite de 1000 règles sur les commutateurs, jusqu'à 128 serveurs peuvent être déployés sans compression. En compressant à la fin, jusqu'à 256 serveurs peuvent être déployés. Enfin, MINNIE permet de déployer jusqu'à 1458 serveurs. Bien que la seule utilisation de la compression permette de déployer plus de 2 fois plus de serveurs, l'utilisation d'un algorithme de routage considérant aussi la capacité des tables permet le déploiement de 6 fois plus de serveurs.

Dans la Table 1, nous observons l'impact de l'utilisation de MINNIE sur plusieurs topologies. Bien que possédant un nombre équivalent de serveurs (environ 1000), ces topologies montrent une disparité flagrante, que ce soit au niveau du nombre de commutateurs ou du nombre de liens. Cela entraîne une différence du nombre moyen et maximum de flots par commutateur présenté dans la deuxième partie du tableau. Les topologies du deuxième groupe montrent un nombre de flots maximum et moyen plus faible que celles du groupe 1. Cette différence vient de la capacité des serveurs à effectuer de la transmission de paquets. Dans la troisième partie du tableau, nous voyons que MINNIE permet de router tous les flots du réseau sans dépasser la limite de 1000 règles sur aucun commutateur. Le taux de compression est d'au minimum 80% et monte jusqu'à plus de 99%. Finalement, le temps de calcul des chemins est en dessous de la milliseconde,

## MINNIE : enfin un monde SDN sans (trop de) règles

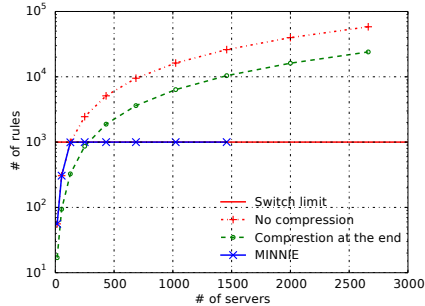


FIGURE 1: Simulation fat tree : nombre maximum de règles en fonction du nombre de serveurs.

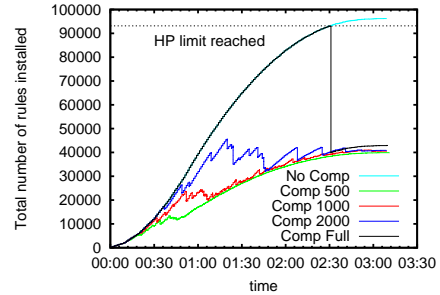


FIGURE 2: Plateforme fat tree : nombre total de règles installées au cours de l'expérimentation.

	Sans comp.	Comp. 500	Comp. 1000	Comp. 2000	Comp. limite
# de compressions	-	16 594	95	28	20
Taux de perte (%)	$6.25 \times 10^{-6}$	0.003	$5.65 \times 10^{-4}$	$2.83 \times 10^{-5}$	$3.7 \times 10^{-4}$

TABLE 2: Nombre total de compressions et taux de perte

et le temps de compression ne dépasse pas les 13ms. L'impact de MINNIE est négligeable sur le délai du réseau, ce que nous confirmons dans la section suivante.

## 3 Plateforme SDN

Nous évaluons maintenant MINNIE sur une plateforme SDN : un  $k = 4$  fat tree composé de 16 serveurs et 20 commutateurs. Ces 20 commutateurs sont émuloés sur un seul commutateur SDN HP 5400zl qui possède une limite totale de 65 536 règles logicielles, partagée entre tous les commutateurs. Chaque serveur héberge 8 machines virtuelles et chaque machine communique avec toutes les autres machines qui ne sont pas hébergées dans le même châssis. Sous ces hypothèses, 67 584 règles sont nécessaires au total afin de pouvoir router tous les flots, ce qui dépasse la limite totale.

Nous considérons trois scénarios distincts : (i) les tables ne sont jamais compressées, (ii) les tables sont individuellement compressées lorsqu'elles atteignent un certain seuil (500, 1000 ou 2000), et (iii) l'ensemble des tables est compressé lorsque la limite totale du nombre de règles est atteinte.

La Figure 2 décrit l'évolution du nombre total de règles installées au cours du temps. Notez que le nombre total de règles comprend aussi des règles installées sur des *Open VSwitch* nécessaire à la détection de nouveaux flots dans le réseau. Pendant la première demi-heure, le nombre de règles est similaire pour tous les scénarios. Chaque baisse du nombre de règles correspond à la compression d'une table. Les seuils à 500 et 1000 règles présentent la progression du nombre de règles la plus faible. En effet, en compressant les tables nous y introduisons des règles d'agrégation, ce qui diminue le nombre de règles à installer lors de l'apparition de nouveaux flots car ceux-ci peuvent appliquer une règle agrégée pré-existante. MINNIE permet ainsi de router tous les flots sans dépasser la limite des 65 536 règles, en offrant des taux de compression compris entre 76% et 97%.

En utilisant un seuil de règles trop bas, le nombre de compressions peut être important, ce qui dégrade le taux de perte. En effet, avec un seuil de 500 règles, plus de 16 000 compressions sont effectuées sur la totalité de l'expérience (voir Table 2). Comme il est impossible de redescendre en dessous du seuil de 500 règles dans la table, à chaque nouvelle règle le module de compression est lancé, ce qui fait monter le taux de perte des paquets à 0.003%. Pour les autres scénarios, le taux de perte est peu affecté par la compression.

La Figure 3 montre le délai de transmission du premier paquet de chaque flot dans le réseau. Comme ce paquet déclenche la détection d'un nouveau flot, son délai comprend le temps de contact avec le contrôleur et le temps d'installation des règles pour le flot. Nous ne comparons ici que les scénarios *sans compression*

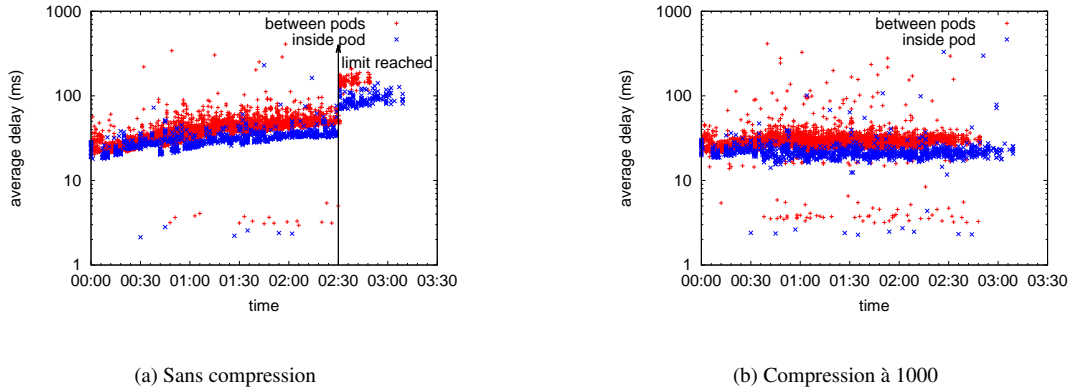


FIGURE 3: Délai du premier paquet

et *compression avec un seuil de 1000 règles*. Sans compression, le délai croît progressivement de 20ms à 60ms alors qu'en compressant régulièrement le délai reste constant, aux alentours de 25 ms. Au bout de 2h30, lorsque la limite des 65 536 règles est atteinte, le délai sans compression bondit au dessus de 100ms, car à chaque saut le contrôleur transmet la règle de "forwarding", mais ne peut la stocker dans la table.

## 4 Conclusion

Dans ce papier, nous présentons MINNIE, un algorithme de routage qui résout le problème des tailles de tables dans les réseaux SDN en agrégeant les règles par sources ou par destinations. Nous montrons par simulation sur une variété d'architectures de centres de données que MINNIE permet de déployer plus d'un million de flots avec seulement 1000 règles au maximum par équipement de transmission. Ces résultats numériques sont confirmés par des expérimentations sur une plateforme SDN. Ils confirment la capacité de MINNIE à réduire le nombre de règles nécessaires sans impact visible sur le délai ni sur le taux de perte.

## Références

- [BK14] S. Banerjee and K. Kannan. Tag-in-tag : Efficient flow table management in sdn switches. In *CNSM*, pages 109–117, 2014.
- [CLENR14] R. Cohen, L. Lewin-Eytan, J.S. Naor, and D. Raz. On the effect of forwarding table size on sdn network utilization. In *INFOCOM*, pages 1734–1742. IEEE, 2014.
- [GHJ<sup>+</sup>09] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta. VL2 : a scalable and flexible data center network. In *SIGCOMM Comp. Commun. Rev.*, volume 39 :4, pages 51–62. ACM, 2009.
- [GHM15] F. Giroire, F. Havet, and J. Moulhierac. Compressing two-dimensional routing tables with order. In *INOC*, pages 1–8, 2015.
- [GMP14] F. Giroire, J. Moulhierac, and T.K. Phan. Optimizing rule placement in software-defined networks for energy-aware routing. In *GLOBECOM*, pages 1–6. IEEE, 2014.
- [MLT12] C.R. Meiners, A.X. Liu, and E. Torng. Bit Weaving : A Non-Prefix Approach to Compressing Packet Classifiers in TCAMs. *IEEE/ACM Trans. on Networking*, 20(2) :488–500, April 2012.
- [nec] NEC univerge PF5240 and PF5820. <http://www.openflow.org/wp/switch-nec/>.
- [RHC<sup>+</sup>16] M. Rifai, N. Huin, C. Caillouet, F. Giroire, J. Moulhierac, D. Lopez Pacheco, and G. Urvoy-Keller. MINNIE : an SDN World with Few Compressed Forwarding Rules. Technical Report RR-8848, Jan. 2016.