De l'utilisation des SDN pour améliorer la fabrique de TouIX

Rémy Lapeyrade, Marc Bruyère, Philippe Owezarski

Rapport sur la première année du projet

De l'utilisation des SDN pour améliorer la fabrique de TouIX

Rémy lapeyrade, Marc Bruyère, Philippe Owezarski

LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

Juillet 2017

Table des matières

1. Introduction

Les nœuds d'échange Internet (ou IXP pour Internet eXchange Point) sont des éléments critiques de l'architecture de l'Internet actuel pour l'interconnexion des réseaux. Leur importance et attrait croissants requièrent des niveaux de flexibilité importants que le concept de réseaux programmables (ou SDN : Software Defined Network) semble en mesure d'offrir. Ensemble, IXP et SDN présentent une promesse pour la mise en place et la gestion de fabriques de communication qui dépasse le traditionnel cadre intra-domaine [1], [2], [3]. En effet, en transportant d'énormes volumes de trafic et en interconnectant de multitudes de réseaux de types différents, les IXPs sont devenus des éléments centraux de l'Internet mondial [4], [5]. Ainsi, les IXPs sont devenus un type de réseaux universel affectant largement le partage des informations dans l'Internet.
Toutefois, transformer les IXPs de leur forme classique en une version complètement SDN représente un challenge scientifique et technologique. La scalabilité et la fiabilité sont deux aspects essentiels des IXPs qui ne peuvent pas être négligés au moment de la migration vers une solution SDN dont l'objectif premier est une plus grande souplesse et flexibilité d'utilisation, couplée à des besoins moindre en matière de ressources humaines. Par exemple, l'interruption des canaux de contrôle vers le plan de données peut provoquer de graves perturbations dans l'IXP, avec de potentielles répercussions sur des centaines de réseaux et des volumes de trafic conséquents [6], [7].
Au cours de la première année du projet TouSIX, nous avons conçu *Umbrella*, une nouvelle approche pour la gestion de la fabrique qui réduit le risque d'une dépendance excessive de la fabrique par rapport au plan de contrôle, et permettant en outre au contrôleur de ne gérer qu'une fabrique aux principes de conception simplifiés. Umbrella s'appuie sur la programmabilité des SDN pour aborder la gestion d'une partie du trafic, le trafic ARP (Address Resolution Protocol), directement dans le plan de données. Dans Umbrella, le seul rôle du contrôleur consiste à superviser le réseau, en s'appuyant sur sa connaissance globale du dit réseau.
Umbrella complète les architectures SDN précédentes pour les IXPs de deux façons : d'abord, Umbrella permet de gérer des IXPs de plus en plus complexes en mettant en œuvre des fabriques plus fiables et plus scalables. Umbrella permet ensuite de mettre en place des architectures SDN-IXP qui ne se limitent pas à des communications mono-saut, ce qui permet d'appliquer cette approche à la plupart des topologies d'IXPs. Nous envisageons les IXP-SDN avec des architectures dans lesquelles le contrôleur agit comme un superviseur intelligent plutôt que comme un élément de décision critique et dangereux. Umbrella est un premier pas dans cette direction.
Globalement, Umbrella fonctionne de la façon suivante : premièrement, le contrôleur Umbrella obtient directement la configuration des membres de l'IXP (association d'adresses MAC et IP, de numéros de ports, …), calcule ses chemins internes à l'IXP et les installe dans tous les commutateurs de l'IXP. Ensuite, pour chaque paquet entrant dans la fabrique, le commutateur OpenFlow (OF) d'entrée encode le numéro de chemin à suivre vers le commutateur de sortie dans le champ de l'adresse MAC de destination. Cela a la particularité par effet de bord de transformer tous les massages broadcast (comme le trafic ARP par exemple) en messages unicast. Les commutateurs de cœur utilisent alors le chemin ainsi encodé pour forwarder les paquets. A la fin, le commutateur de sortie rétablit l'adresse MAC de destination dans le champ approprié du paquet.

Les principales contributions des travaux effectués lors de la première année du projet TouSIX sont :

- Nous avons proposé l'architecture Umbrella et montré comment elle influence la programmabilité SDN dans le plan de données. Nous avons aussi montré comment déployer de façon incrémentale Umbrella, et démontré ses aspects pratiques au travers de son déploiement réel sur le réseau TouIX.
- Nous avons montré comment Umbrella étend les solutions actuelles pour les IXP-SDN. Umbrella est compatible avec les solutions actuelles [3] et permet leur implémentation sur des IXPs multi-sauts, tout en réduisant les risques de disfonctionnement du plan de données.
- Finalement, nous diffusons notre implémentation d'Umbrella, et notamment de son application TouSIX-manager qui génère les règles Umbrella et la configuration BIRD pour le server de route (ou RS pour Route server).

2. Contexte et motivation

2.1. L'environnement d'un IXP

Les IXPs sont des fabriques d'interconnexion où se croisent de nombreux réseaux (i.e. les membres) pour échanger des grandes quantités de trafic [8]. Les IXPs sont classiquement implémentés comme des domaines broadcast de niveau 2 auxquels les membres connectent des routeurs et échangent du trafic. Ces routeurs annoncent leurs routes via le protocole BGP. Pour faciliter les peerings multi-latéraux, les IXPs installent et opèrent des serveurs de routes (RS) [9], [10].

Le réseau d'un IXP se compose de commutateurs de bordure et de cœur [8]. Les commutateurs de bordure se connectent aux routeurs des membres, et les commutateurs de cœur interconnectent les différentes localisations physiques le l'IXP et agrègent le trafic de l'IXP. Alors que certaines topologies d'IXPs n'implémentent que des solutions mono-saut, i.e. les paquets ne traversent qu'un seul commutateur de cœur (e.g. AMS-IX et DE-CIX), d'autres permettent d'avoir plusieurs sauts dans leur cœur (e.g. LINX et MSK-IX).

Les IXPs opèrent généralement au niveau Ethernet et sont rarement impliqués dans les décisions de routage. Ces fabriques Ethernet sont cependant sujettes à des erreurs, à cause de possibles pannes de composants du réseau et/ou des boucles dans le réseau. Les systèmes utilisant le protocole de Spanning tree ou les architectures MPLS ne sont que des solutions partielles à ces problèmes. Pour réduire le bruit dû au trafic non sollicité sur le domaine broadcast et éliminer les boucles Ethernet, les IXPs peuvent exploiter la nature statique de l'ensemble des objets connectés et filtrer leur trafic sur la base des adresses MAC. Ces filtres garantissent que les ports des routeurs des membres n'acceptent que le trafic qui contient l'adresse MAC des routeurs des membres connectés, ce qui ne change que lorsque de nouveaux routeurs sont installés ou désinstallés par les membres. En particulier, les IXPs utilisent le protocole ARP dans leur plan de contrôle pour associer les adresses IP des membres avec l'adresse MAC de leurs routeurs.

Le trafic ARP dans les fabriques des gros IXPs peut être suffisant pour compromettre les routeurs de membres ayant des capacités faibles [11]. De plus, le trafic ARP peut compromettre le canal de contrôle ce qui conduit à de graves perturbations dans l'IXP [6], [7]. La quantité de trafic ARP est d'autant plus importante dans le cas de panne du réseau [12],

lorsque de nombreux routeurs essaient de récupérer l'adresse IP des pairs alors que ces adresses sont indisponibles du fait de la panne. L'état de l'art pour résoudre ce genre de problème lié à un excès de trafic ARP se résume aujourd'hui à mettre en œuvre une solution de type ARP-Sponge[1]

ARP-Sponge: Un serveur ARP-Sponge limite le trafic ARP en deçà d'un seuil donné. Lorsque le nombre de requêtes ARP pour une adresse IP donnée atteint ce seuil (par exemple parce qu'une interface ne répond plus), le serveur ARP-sponge « éponge » l'adresse IP en question : le serveur répond avec sa propre adresse MAC à la requête ARP du nœud, et de là tout le trafic ARP envoyé au nœud est transmis au serveur ARP-Sponge. Lorsque le serveur ARP-Sponge reçoit du trafic d'une adresse IP « épongée », il cesse de l'éponger.

Toutefois, même si le mécanisme ARP-Sponge élimine le risque de surcharge en terme de trafic ARP, il présente plusieurs limites :

• Son unicité en fait un point de panne sans solution de secours. Et si on introduit plusieurs serveurs ARP-Sponge fonctionnant en parallèle, cela contribue à grandement complexifier la fabrique.

• Les serveurs ARP-Sponge n'éliminent pas tout le trafic ARP non voulu.

• Les serveurs ARP-Sponge utilisent des heuristiques pour déterminer si une adresse IP est de nouveau accessible. En particulier, cela nécessite de pouvoir diffuser des messages en mode broadcast, ce qui consomme des ressources au niveau de tous les routeurs connectés.

• Lorsqu'une interface est de nouveau active, le serveur ARP-Sponge peut ne pas le détecter. Le serveur ARP-Sponge doit donc sonder périodiquement la fabrique à l'aide de messages ARP pour détecter les adresses « épongées ». Si une réponse est reçue, il retire de la liste des adresses épongées celle qui vient de répondre. Il peut arriver cependant que certains équipements ne soient pas sondés et que leur retour en activité ne soit pas détecté.

Route Servers (RS): A la base, pour que deux membres d'un IXP qui veulent échanger du trafic au travers de l'IXP puissent le faire, ils doivent établir une session BGP bi-latérale au travers du réseau de l'IXP. Chaque nœud BGP connecté à la fabrique doit donc établir des sessions BGP avec tous les autres membres de l'IXP afin d'en obtenir des informations sur les préfixes d'adresses de l'IXP atteignables. Comme les IXPs deviennent de plus en plus grands [13], cette solution s'avère inapplicable car elle nécessiterait de maintenir un très grand nombre de sessions BGP. Cela engendrerait une charge d'administration importante et des risques de saturation des routeurs des membres. Les IXPs ont donc introduit dans leurs réseaux des serveurs de routes (RS), ce qui offre une réelle valeur ajoutée pour leurs membres [10]. Les RSs stockent toutes les informations sur les routes entrantes fournies par les membres et les diffusent sans modification aux autres membres. Grâce au RS, un membre d'IXP peut recevoir toutes les informations de routage disponibles pour l'IXP grâce à une session BGP unique.

[1] Manuel sur ARP-sponge : http://ams-ix.net/downloads/arpsponge

2.2. Séparation forte des plans de contrôle et de données

De précédent travaux ont déjà montré qu'OpenFlow (OF) [14] peut être utilisé avec succès au niveau de nœuds d'échange [15], [16], [2], [3]. Ces contributions considèrent une fabrique d'IXP avec un contrôleur central en guise de plan de contrôle pour tous les routeurs de l'IXP. Avec une telle architecture, il est légitime de co-localiser le contrôleur et le serveur de routes pour s'assurer que les deux plans de contrôle SDN et BGP peuvent communiquer aisément et avec de faibles délais [2], [3]. La Figure 1 illustre un exemple d'IXP utilisant une approche SDN dans lequel tous les messages de contrôle échangés entre les routeurs sont transmis au contrôleur SDN, qui agit ensuite en leur nom et programme en conséquence les commutateurs de la fabrique.

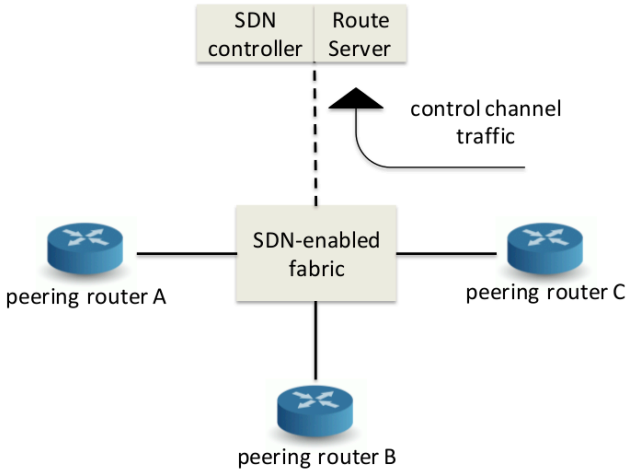

Figure 1. SDN-IXP: un cas d'étude

Nous avons émulé ce scénario sur Mininext [17], une couche d'extension pour bâtir des scénarios complexes au dessus de Mininet. Nous avons instancié deux types de conteneurs virtuels pour les routeurs de bordure, un pour le RS et deux autres pour les machines des clients directement connectés à un de ces deux routeurs. Nous avons émulé l'IXP par un unique contrôleur open vSwich couplé à Ryu [18], et agissant également comme un proxy ARP comme cela est suggéré dans [15], [16], [2], [3].

Malgré l'aspect intéressant d'une telle architecture, e.g., permettant des polices plus riches, nous avons découvert qu'un des problème reste, à savoir le couplage entre les plans de contrôle et de données.

En effet, les problèmes du plan de données peuvent affecter les messages du plan de contrôle, et par conséquent entrainer des pertes de performance du plan de contrôle, aggravant encore plus l'effet sur le plan de données. Le problème critique se situe au niveau du proxy-ARP centralisé : si le trafic dans le canal de contrôle est, pour une raison ou une autre, touché par une augmentation de délai, tous les mécanismes en mode connecté (i.e., BGP, TCP) peuvent dysfonctionner. Par exemple, imaginons que des messages ARP d'un routeur de bordure A soient retardés dans la fabrique de l'IXP lors de l'accès au contrôleur SDN : cela impactera toutes les sessions BGP entre le routeur A et ses pairs, provoquant ainsi (dans le pire des scénarios) l'établissement de nouvelles connexions. La Figure 2 montre la période temporelle durant laquelle le plan de données ou BGP dysfonctionneraient si les massages ARP subissaient un certain retard. La Figure 2 montre que même un court délai de quelques

millisecondes pour les messages ARP peut provoquer des dysfonctionnements sur le plan de données qui apparaissent dans des proportions bien plus importantes. Etant donnés les volumes de trafic échangés sur la fabrique de l'IXP, de tels dysfonctionnements du plan de données ne sont pas acceptables [6], [7]. Pour bénéficier complètement des avancées proposées dans [15], [16], [2], [3], nous allons mettre en place une séparation encore plus grande entre les plans de contrôle et de données. L'approche que nous avons proposée est l'une des solutions possibles pour la mise en place d'un tel niveau de séparation.

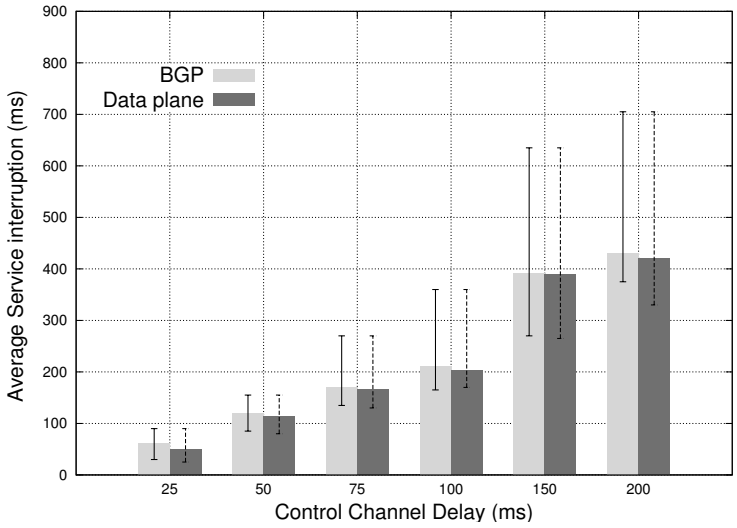

Figure 2. Dépendance entre les plans de contrôle et de données

3. Une nouvelle fabrique SDN pour les IXPs

Cette partie a pour objectif de présenter Umbrella, une nouvelle fabrique SDN pour les IXPs qui se focalise sur l'indépendance entre les plans de contrôle et de données pour fournir des transferts de données fiables, robustes et scalables au sein de l'IXP.

3.1. Pas de trafic en mode broadcast

Les IXPs appliquent des règles strictes [19], [20] pour réduire l'effet de bord dû à l'utilisation d'un domaine broadcast partagé de niveau 2 ; par exemple, l'adresse MAC du routeur sur lequel les membres se connectent à la fabrique doit être connu à l'avance. C'est seulement à ce moment que l'IXP alloue un port Ethernet sur le commutateur de bordure et une adresse IP publique de son espace [21] et configure une liste de contrôle d'accès (ou ACL pour Access control List) avec cette adresse MAC. C'est ainsi que la localisation de tous les routeurs des membres est connue de l'IXP. C'est pour cela qu'Umbrella élimine le besoin d'un mécanisme de découverte des localisations des équipements et qui reposerait sur des messages en mode broadcast (i.e., requêtes ARP, IPv6 neighbor discovery) et par conséquent rend inutile un proxy-ARP comme cela est proposé dans les précédentes solutions pour les IXP-SDN [15], [16], [2], [3]. Umbrella convertit à la volée les paquets broadcast en paquets unicast en utilisant la possibilité offerte par OF de ré-écrire le champ d'adresse MAC et qui correspond à une règle donnée [22]. La Figure 3 montre une vision de haut niveau de l'idée proposée en utilisant la topologie d'IXP représentée sur la Figure 4 comme scénario de référence.

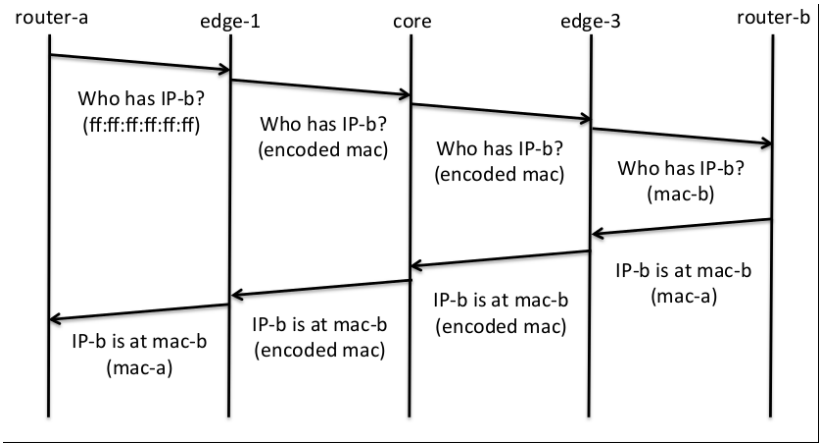

Figure 3. Handshake pour les requêtes ARP

Nous proposons d'utiliser un mécanisme de forwarding basé sur des labels pour réduire le nombre de règles dans le cœur de la fabrique de l'IXP. Les commutateurs de bordure Umbrella écrivent explicitement les ports destination pour chaque saut dans le champ destination de la trame MAC du paquet. Le premier octet de l'adresse MAC indique donc pour chacun des commutateurs de cœur traversés le numéro de port de sortie à utiliser.

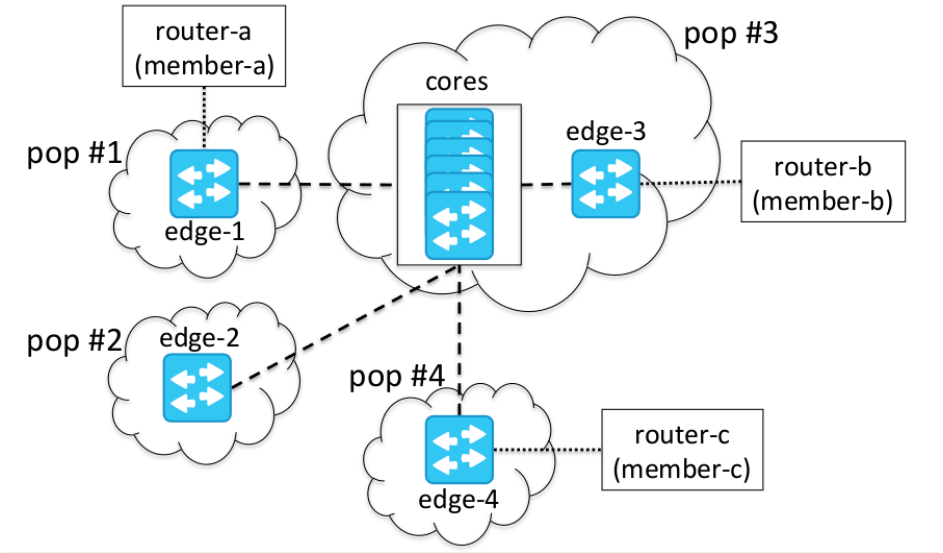

Figure 4. Topologie classique d'un IXP de taille significative

Le tableau 1montre un exemple d'une table de flux d'un commutateur de cœur d'un IXP qui utilise l'approche Umbrella. Avec Umbrella, le nombre d'entrées dans la table des flux par commutateur de cœur reste limité par rapport au nombre de ports physiques actifs dans le commutateur. Cet aspect est important pour garantir la scalabilité de la fabrique. Le mécanisme d'encodage d'Umbrella est aujourd'hui limité à 256 ports de sortie par saut. Toutefois, il est tout à fait possible d'utiliser plus de bits si ce nombre n'est pas suffisant.

Adresse MAC de destination (Masque)	Port de sortie
01 :00 :00 :00 :00 :00 (ff :00 :00 :00 :00 :00	1
02 :00 :00 :00 :00 :00 (ff :00 :00 :00 :00 :00	2
03 :00 :00 :00 :00 :00 (ff :00 :00 :00 :00 :00	3
04 :00 :00 :00 :00 :00 (ff :00 :00 :00 :00 :00	4

Tableau 1. Exemple d'une table de flux Umbrella dans un commutateur de cœur d'un IXP

Nous expliquons maintenant comment Umbrella fonctionne sur la topologie décrite sur la Figure 4. Pour deux membres a et b qui veulent établir une association (peering), Umbrella détermine le chemin au sein de la fabrique de l'IXP de la façon suivante. Le commutateur de bordure edge-3 est connecté à un commutateur de cœur par le port 2 et à router-b par le port 3. Le routeur router-a du membre member-a envoie une requête ARP (i.e., un message broadcast) au routeur router-b. Le commutateur edge-1 reçoit la trame, ré-écrit la trame avec le champ d'adresse MAC de destination avec les ports correspondants, 2 et 3, 02:03:00:00:00:00, et le forwarde au commutateur de cœur indiqué. Lorsque la trame atteint le cœur, elle est redirigée vers le port de sortie 2, et ensuite vers le commutateur edge-3 (i.e., le forwarding dans le cœur repose sur l'octet de poids fort). Enfin, edge-3, avant de forwarder la trame par le port de sortie indiqué dans le second octet de l'adresse MAC, ré-écrit ce champs avec l'adresse réelle du routeur router-b.

Lorsque la source et la destination sont directement connectées au même commutateur de bordure, aucun encodage n'est requis, et l'adresse broadcast de destination est directement remplacée par l'adresse MAC de destination par le commutateur de bordure concerné. Dans un scénario IPv6, l'indication OF dans le commutateur de bordure doit être placée dans le champs *IPv6 ND target* du paquet de sollicitation *ICMPv6 Neighbor* [23]. La table de correspondance sur le commutateur de bordure doit conserver les associations entre les adresses IPv6 et leur localisation, comme dans le cas IP4.

3.2. Une approche de commutation basée sur des labels

Le mécanisme de propagation (forwarding) d'umbrella permet d'utiliser des commutateurs traditionnels (sans implémentation d'OF) dans le cœur, limitant ainsi les investissements (et coûts) de mise à jours des matériels. Un commutateur de cœur doit seulement forwarder les paquets sur la base de règles de filtrage d'accès, alors que les commutateurs de bordure doivent intégrer OF pour ré-écrire le champ MAC de destination.

Cette approche est directement applicable pour des IXPs permettant un seul saut dans le cœur (comme AMS-IX et DE-CIX), mais n'est pas applicable pour des fabriques autorisant plusieurs sauts (comme LINX et MSK-IX). Avec un seul saut, le port de sortie est encodé dans l'octet de poids fort de l'adresse MAC de destination. Dans le cas multi-sauts, comme un paquet peut traverser plusieurs commutateurs de cœur, un nouveau mécanisme d'encodage est requis pour indiquer les différents ports de sortie au niveau des différents commutateurs de cœur.

La figure 5 présente un exemple de fabrique d'IXP à sauts multiples dans le cœur. Le commutateur de bordure edge-a doit traverser deux commutateurs de cœur sur le chemin b pour atteindre edge-d. C'est un cas très fréquent dans les topologies hypercube, comme celles adoptées par LINX ou MSK-IX.

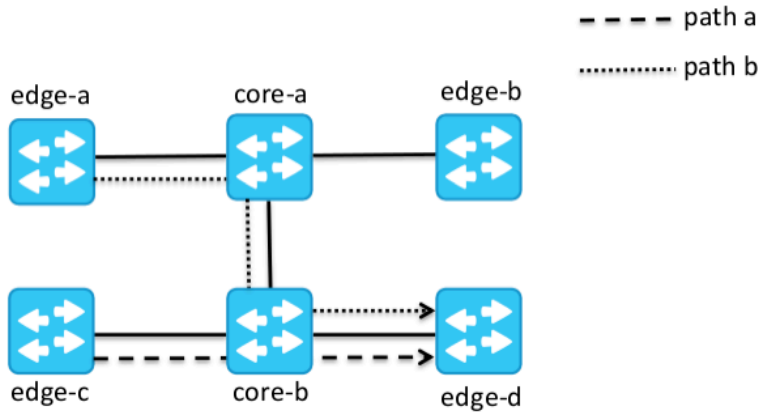

Figure 5. Exemple de cœur multi-sauts

Adapter Umbrella aux topologies multi-sauts d'IXP n'est pas trivial. Il faudrait pour cela définir un mécanisme d'encodage des adresses MAC de destination dans lequel l'octet de poids fort indique le port de sortie du premier commutateur de cœur (i.e., core-a), le second octet indique le port de sortie du second commutateur de cœur (i.e., core-b), et ainsi de suite. C'est tout bonnement impossible. En effet, selon la route, un commutateur de cœur peut être soit le premier soit le second sur le chemin à suivre. Une autre solution pourrait consister à considérer les ports d'entrée de la trame dans les règles de forwarding installées au niveau des commutateurs de cœur. Avec les ports d'entrée, on peut savoir où se situe le commutateur sur le chemin entre la source et la destination, et ainsi considérer le bon octet dans le champ de l'adresse MAC de destination. Hélas, cette approche ne peut pas fonctionner sur des topologies arbitraires. De plus, un tel mécanisme conduirait à une explosion du nombre de règles dans le cœur, le nombre de règles de forwarding grandissant de façon quadratique avec le nombre de ports d'entrée possibles.

En mettant en place une méthode à base de *source routing*, Umbrella s'absout enfin de ce problème. Dans l'approche Umbrella pour les fabriques multi-sauts d'IXP, le premier commutateur de bordure doit sélectionner le chemin à emprunter pour chaque paquet. Une liste ordonnée des ports de sortie est alors encodée dans le champ d'adresse MAC de destination, comme une pile de labels. Finalement, chaque nœud de cœur traite les trames en considérant la valeur au sommet de la pile, et la dépile avant de les forwarder (Figure 6).

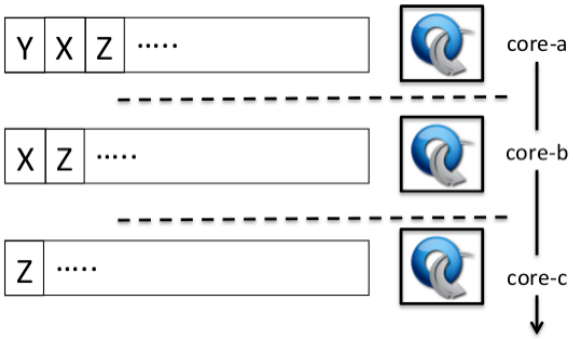

Figure 6. Exemple pour le mécanisme de forwarding d'Umbrella

Avec cette configuration, chaque commutateur n'a besoin de ne regarder que l'octet de poids fort de l'adresse, peut importe la place qu'il occupe sur le chemin vers la destination. Retirer de la pile le dernier label utilisé nécessite de pouvoir ré-écrire les entêtes, rendant cette solution possible uniquement sur des commutateurs de cœur compatibles avec OF. En particulier, chaque commutateur de cœur doit posséder deux tables d'actions : forwarding et *coppy-field*[2]. Cette solution présente donc deux limites pratiques (qui seront détaillées et traiter plus longuement dans la partie 5.2) :

- Le nombre maximum de ports de sorties qui peuvent être adressés lors de chaque saut est limité à 256, puisque le numéro de port de sortie pour chaque commutateur est encodé dans l'octet de poids fort du champ d'adresse MAC de destination.
- Le nombre maximal de sauts au sein de l'IXP est limité à 6, car nous ne pouvons utiliser que les 6 octets de l'adresse MAC pour encoder tout le cheminement des trames.

Cependant, nous devons surtout considérer Umbrella comme une approche permettant une séparation plus forte entre les plans de contrôle et de données. Les limitations pratiques peuvent se résoudre avec l'apparition de nouvelles implémentations des différents protocoles.

3.3. Umbrella et les serveurs de routes

Umbrella peut traiter le forwarding du trafic BGP au sein de l'IXP, à la fois pour les peerings bi-latéraux et les RSs. Pour les sessions BGP bi-latérales, la connexion TCP est traitée comme du trafic traversant l'IXP au niveau du plan de données, et le trafic est traité selon les règles du commutateur, et sans intervention du plan de contrôle. Pour les serveurs de routes, le trafic BGP entrant dans la fabrique est dirigé vers le RS grâce à une règle simple au niveau du commutateur de bordure, alors que le trafic BGP sortant est traité selon les règles classiques de propagation établies au niveau des commutateurs de bordure.

3.4. Détection de pannes et récupération

- [2] Les specifications d'OF 1.5 intègrent ces capacités de copie et écriture des champs des entêtes.

Group fast failover est le mécanisme implémenté dans OF 1.1 pour réagir en cas de panne sur des liens. Une table *fast failover group* peut être configurée pour monitorer le statut des ports, des interfaces et des actions de forwarding des commutateurs, indépendamment du contrôleur. Récupérer d'une panne du plan de donnée est plus délicate. Dans un tel scénario, le contrôleur doit sonder (activement) l'état du plan de données. En particulier, le contrôleur Umbrella peut avoir à implémenter le protocole LLDP (Local Link Discovery Protocol) [24], ou le protocole BFD (Bidirectional Forwarding Detection) [25]. Une fois que la panne du plan de données a été détectée, le contrôleur Umbrella change seulement la configuration des commutateurs de bordure avec un chemin *fallback*.

4. Principaux avantages

Umbrella a été conçue pour être au maximum flexible. Cela peut se faire de manière neutre au niveau de la couche 3 ou au niveau service (i.e., application), selon les paramètres de configuration choisis. Cette partie énonce les avantages apportés par l'utilisation d'Umbrella pour les IXPs.

Scalabilité & fiabilité

Umbrella améliore la scalabilité globale de la fabrique car elle ne nécessite qu'un petit nombre de règles au niveau des commutateurs de cœur et bordure (cela est démontré dans la partie 5). De plus, cela évite au contrôleur de traiter les paquets de découverte de la localisation, et par conséquent réduit la dépendance entre les plans données et contrôle. Plus avantageux encore, cela permet au plan de contrôle de continuer à fonctionner même si le niveau de performance du canal de contrôle est faible.

IXP orientés services

La commutation orientée labels des mécanismes de forwarding rend la fabrique de l'IXP orientée services. Un IXP orienté service peut créer des catalogues de ressources réseaux avec leurs politiques de gestion associées (e.g., paramètres de QoS et bande passante) qui peuvent être appliquées aux applications actives dans le réseau. Comme les chemins dans la fabrique de commutation sont configurés au niveau du commutateur de bordure, il est possible de configurer différents chemins selon les applications, ou de rediriger certains flux sur différents chemins suivant les services activés (e.g., firewal, QoS ou monitoring). Notons que c'est seulement une facilité qui peut être activée si besoin. Umbrella peut évidemment être utilisée de façon neutre au niveau de la couche 3 (et au dessus), comme cela est fait traditionnellement aujourd'hui dans les réseaux d'IXP.

Compatibilité avec les commutateurs traditionnels

Si la topologie de l'IXP ne présente qu'un seul saut au niveau de son cœur, n'importe quel commutateur pouvant réaliser le routage sur la base de listes d'accès au niveau MAC peut être

utilisé dans le cœur avec Umbrella (comme vu dans la partie précédente). En effet, aucune autre opération que l'association et le forwarding des paquets sur la base de masques de niveau 2 n'est réalisée au niveau du cœur du réseau, ce qui rend cette architecture compatible même pour des commutateurs non OpenFlow.

Nature pseudo-câble

Un pseudo-câble[3] est une émulation d'une connexion point à point sur un réseau à commutation de paquets. Comme présenté plus haut, avec Umbrella, tout le trafic broadcast (ARP IPv4 et ICMPv6 ND) est converti en unicast en bordure du réseau, résolvant ainsi les problèmes liés à l'utilisation d'un domaine de diffusion partagé. Umbrella garantit que chacun des membres ne reçoit que le trafic qu'il est supposé pouvoir voir. Cela résout également les problèmes de traitements processeur importants en bordure du réseau pour la détection et l'analyse du trafic non souhaité (parmi lequel on trouve les paquets broadcastés).

Visibilité

Dans Umbrella, le chemin réel à emprunter pour les paquets est encodé à la place de l'adresse MAC de destination. Cela implique une visibilité totale des chemins de forwarding au cœur de la fabrique de l'IXP, ce qui peut être exploité pour améliorer les mécanismes de récupération de panne, et plus généralement pour la gestion générale du réseau de l'IXP.

5. Evaluation expérimentale

Dans cette partie, nous évaluons différents aspects de l'architecture Umbrella. Nous commençons par estimer le nombre moyen de règles pour le traitement des flux au niveau des commutateurs de bordure, et ce pour différentes tailles de fabriques d'IXP (partie 5.1). Nous étudions ensuite l'applicabilité de l'approche de commutation par labels dans des scénarios réels (partie 5.2). Enfin, nous estimons l'impact du proxy ARP sur les performances du plan de données, et comparons les résultats avec l'approche Umbrella (partie 5.3).

5.1. Dimension de la table de règles de flux en bordure du réseau

Associer l'adresse MAC de destination au chemin à l'intérieur de la fabrique requiert de nouvelles règles en bordure de cette même fabrique. Cette partie analyse la scalabilité du mécanisme en terme de nombres de règles requises. Au niveau d'un point d'entrée (ingress), Umbrella a besoin que le commutateur OS en bordure stocke trois règles d'entrées par routeur pair connecté à la fabrique. C'est une condition nécessaire et suffisante pour permettre le routage dans n'importe quelle situation (i.e., IPv4 ARP, IPv6 Neighbor Solicitation et le trafic

[3] http://tools.ietf.org/html/rfc3985

du plan de données). Ce qui suit est un échantillon des trois entrées relatives à un routeur pair :

“eth type”:0x806, “arp op”:1, “arp tpa”:“1.1.1.3”, “actions”:[“type”:“SET FIELD”,“field”:“eth dst”, “value”:“18:01:00:00:00:00”, “type”:“OUTPUT”, “port”:“49”]

“eth type”:0x86DD, “ip proto”:58, “icmpv6 type”:135, “ipv6 nd target”:“2001::1/128”, “actions”:[“type”:“SET FIELD”, “field”:“eth dst”, “value”:“18:01:00:00:00:00”, “type”:“OUTPUT”, “port”:“49”]

“dl dst”:“00:00:00:00:00:03”, “actions”:[“type”:“SET FIELD”, “field”:“eth dst”, “value”:“18:01:00:00:00:00”, “type”:“OUTPUT”, “port”:“49”]

La première entrée concerne le trafic IPv4 ARP entrant. Il est important de tenir compte de l’adresse IP du routeur pair vers lequel est dirigé le trafic de découverte de la localisation. Le champ d’adresse destination MAC (i.e., broadcast) sera ré-écrite avec un chemin prédéfini au cœur de la fabrique. Les seconde et troisième entrées concernent le trafic IPv6 Neighbor Solicitation et le trafic du plan de données. A noter que ces trois entrées partagent une action commune dans les processus de ré-écriture et de forwarding car les traitements d’Umbrella sont communs dans ces deux cas. Sur l’exemple, c’est donc le commutateur de cœur qui va forwarder le trafic sur le port 18, et le commutateur de sortie (egress) l’envoie sur le port 01. Au niveau du trafic sortant, chaque commutateur de bordure doit ré-écrire le champ d’adresse MAC de destination de la trame reçue avec la bonne cible. Comme la valeur à écrire dans le champs d’adresse MAC de destination dépend du port de sortie du commutateur de bordure, le nombre de règles dépend du nombre de routeurs pairs connectés à ce commutateur de bordure. Le nombre de règles pour un commutateur de bordure est ainsi la somme des nombres de règles d’entrées et de règles de sortie.

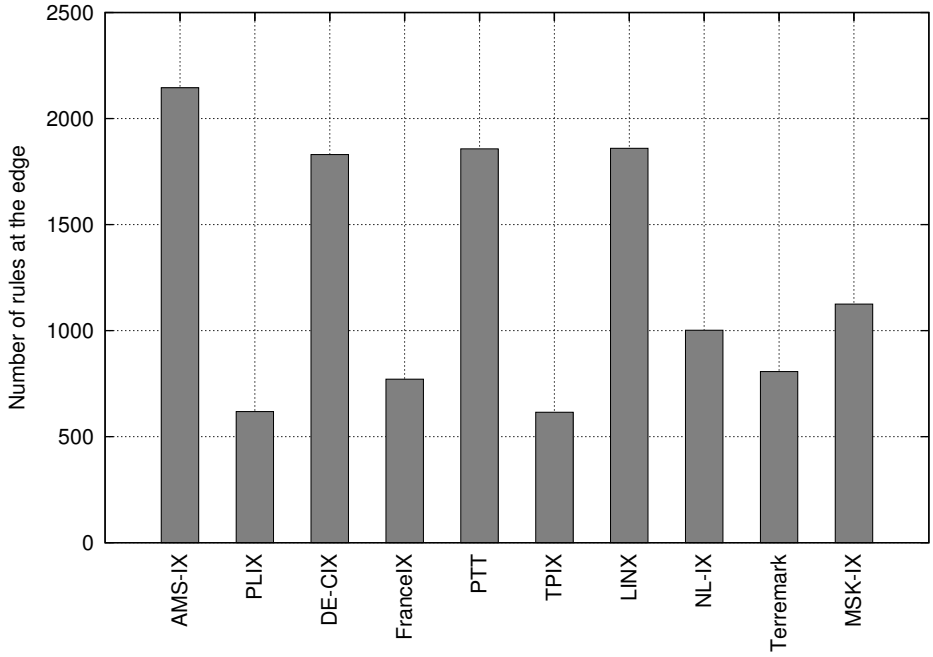

Figure 7. Nombre moyen de règles requises en bordure.

La Figure 7 montre le nombre moyen de règles par commutateur de bordure si Umbrella est implémenté sur différents IXPs européens (en mode neutre de niveau 3). Comme le nombre de règles dépend du nombre de routeurs pairs connectés à la fabrique, nous avons utilisé PeeringDB pour valider cette étude [26], [27].

L'architecture Umbrella est utilisable dans les IXPs actuels, comme le montre le petit nombre de règles indiquées sur la Figure 7. En effet, les commutateurs OF peuvent aujourd'hui fonctionner avec des nombres de règles allant de quelques milliers (e.g., les commutateurs Pica8[4]) à des centaines de milliers (e.g., commutateurs Corsa[5] et Noviflow[6]).

5.2. Nombre de sauts dans la fabrique de l'IXP

Nous avons déjà mentionné une des limitations pratiques de notre approche dans la partie 3.2 : le nombre maximal de sauts au sein de l'IXP ne peut pas dépasser 6, de par notre utilisation du champ d'adresse MAC de destination pour enregistrer le cheminement de la trame jusqu'à son point de sortie (8 bits par saut). Alors que cette contrainte est typiquement vérifiée par conception au niveau des fabriques actuelles avec un seul saut dans le cœur, cela peut ne pas être le cas dans les fabriques d'IXPs multi-sauts, notamment lors des procédures de restauration après des pannes de liens, qui engendre pour les trames de suivre des chemins

[4] http://pica8.org/blogs/?p=565

[5] http://www.corsa.com/sdn-done-right/

[6] http://noviflow.com

plus longs. Nous avons donc étudié les topologies des fabriques des quatre plus grands IXPs européens (quand elles sont disponibles publiquement), et estimé le nombre maximal de sauts qu'un paquet peut avoir à effectuer dans le pire des cas.

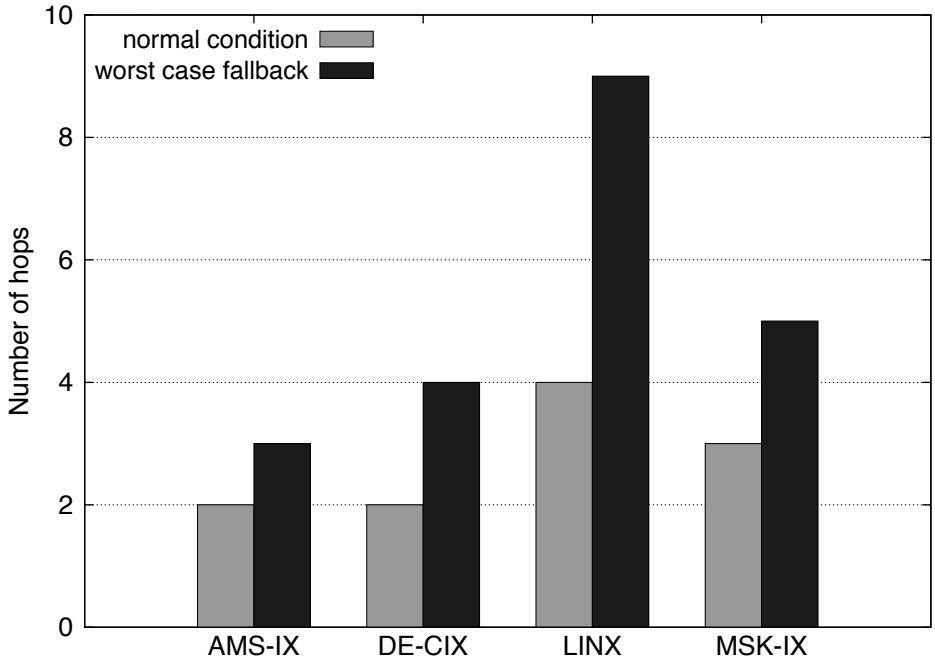

Fig. 8. Nombre de sauts moyens et maximaux dans les principaux IXPs européens

La Figure 8 montre pour chaque fabrique le nombre maximum de sauts lors de cas d'utilisation normaux, mais aussi en cas de panne massive de liens. On définit une panne massive de liens comme le nombre maximal de pannes de lien que la fabrique peut supporter sans dégrader la connectivité entre tous les routeurs pairs. Les résultats pour AMS-IX et DE-CIX lors de leur fonctionnement normal sont conformes à nos attentes en n'autorisant qu'un seul saut. La différence majeure entre les deux fabriques se manifeste en cas de panne massive de liens. Alors que AMS-IX duplique les routeurs de cœur (conduisant à un maximum de trois sauts), DE-CIX utilise quatre commutateurs de cœur connectés selon une topologie totalement maillée (conduisant à un maximum de quatre sauts). Même si nos besoins ne sont pas remplis dans le cas de LINX dans le cas de restauration suite à une restauration/récupération après panne, nous pensons que ce n'est pas réellement une limitation pour Umbrella : de telles pannes perturberaient grandement le trafic, conduisant à des congestions fatales dans la fabrique. La topologie LINX peut être vue comme un hypercube où chaque sommet est un commutateur de cœur. Le nombre maximal de sauts au cours d'une panne massive de liens peut par conséquent impliquer 9 liens. Si nécessaire, une telle situation devrait plutôt être traitée par la mise en place de mécanismes de protection, e.g., des chemins MPLS pré-établis, les chemins pour le trafic dans la partie restante de la fabrique devant être choisis avec soin.

5.3. Impact de canaux de contrôle non fiables sur la performance du plan de données

Pour estimer l'impact d'un canal de contrôle non fiable sur la performance du plan de données, et ainsi évaluer les bénéfices relatifs à l'utilisation d'Umbrella, nous avons mené une campagne de test divisée en trois phases distinctes : nous avons d'abord comparé les effets de l'introduction d'un délai artificiel sur une connexion point à point entre deux machines pour une solution utilisant un proxy ARP d'un côté et Umbrella de l'autre. Ensuite, nous avons étudié les effets de pertes de paquets avec le même scénario. Enfin, nous avons réalisé une longue émulation d'une topologie plus conséquente pour comprendre ces effets sur un scénario plus réaliste.

1) Impact du délai dans le canal de contrôle*:* En utilisant Mininext, nous avons émulé un scénario intégrant deux hôtes interconnectés par un Open vSwitch. Nous avons pré-installé deux règles de flux pour permettre les communications bi-directionnelles entre les deux hôtes, et une autre règle pour rediriger les paquets ARP vers le contrôleur Ryu, sur lequel fonctionne le proxy ARP. La Figure 9 montre le RTT (Round Trip Time) médian des paquets ARP (le temps entre l'émission de la requête ARP et la réception de sa réponse) quand le cache ARP de l'émetteur est vide et que le canal de contrôle subit des délais variés. La Figure 9 montre que tout délai dans le plan de contrôle conduit à de longs délais qui sont plusieurs fois plus importants que le délai initial. Cela montre la relation forte entre les entrées du cache ARP et la performance du plan ce contrôle lorsqu'une architecture SDN repose sur un proxy-ARP.

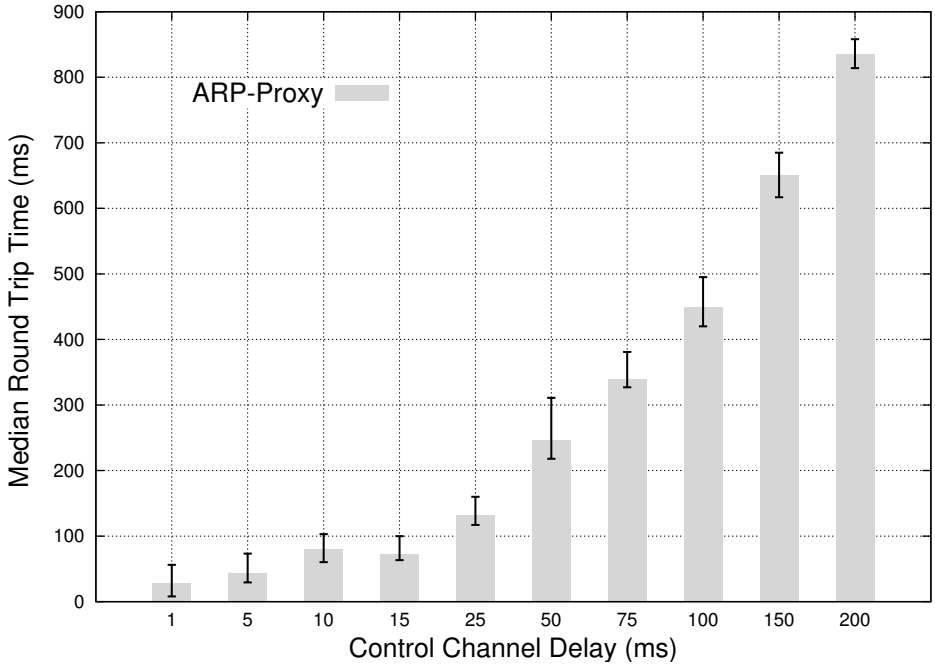

Figure 9. RTT d'un paquet ARP lorsque le proxy-ARP est actif, le cache ARP est vide, et le canal de contrôle subit des délais variables.

La Figure 10 montre que, comme attendu du fait de la conception d'Umbrella, avec Umbrella le plan de contrôle n'a aucun effet dur le RTT des paquets ARP. Notons que le test est aussi effectué avec un cache ARP vide au niveau de l'émetteur. Avec Umbrella, la requête ARP

émise par le hôte est transmise par le Open vSwitch au hôte de réception qui émet ensuite la réponse directement au hôte initial, sans passer par un proxy-ARP.

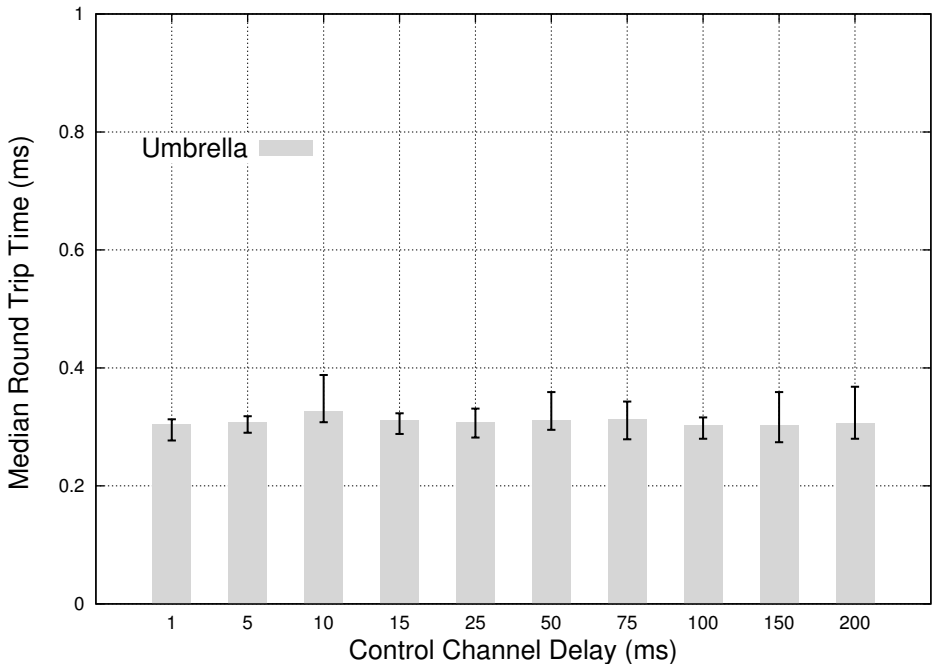

Figure 10. RTT d'un paquet ARP avec Umbrella, le cache ARP est vide et le canal de contrôle subit des délais variables.

Comme les IXPs sont de plus en plus nombreux [28], de plus en plus gros [13], et de plus en plus adaptés [8], [29], il est important de réduire autant que possible les dépendances dangereuses entre les plans de contrôle et de données, notamment quand le cache ARP d'un routeur pair se remplit ou si une entrée expire. La Figure 11 montre le ratio entre le délai introduit sur les paquets ARP et le délai introduit sur le canal de contrôle. Nous observons que quand le délai augmente de plus de 50ms sur le canal de contrôle, le ratio augmente de plus de 10 fois si un proxy-ARP est utilisé. D'un autre côté, Umbrella est complètement insensible aux délais du plan de contrôle, et a un ratio très proche de 1 pour toutes les valeurs de délai.

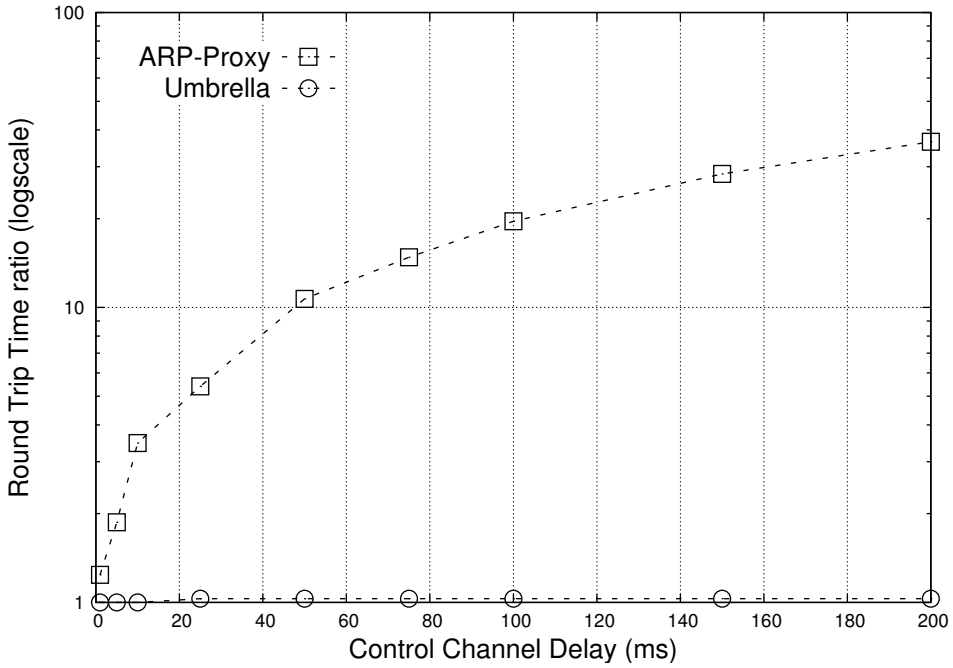

Figure 11. Ratio entre le RTT de paquets ARP et le délai introduit au niveau du canal de contrôle

2) Perte de paquets sur le canal de contrôle *:* L'analyse des délais sur le canal de contrôle a d'ores et déjà démontré les bénéfices de l'utilisation d'Umbrella. Nous allons maintenant étudier l'impact des pertes sur le canal de contrôle sur le RTT des paquets ARP. Les Figures 12 et 13 montrent le RTT médian pour des approches proxy-ARP et Umbrella respectivement.

La Figure 13 met en évidence la forte séparation entre les plans de contrôle et de données avec Umbrella : il apparaît une insensibilité presque totale entre le taux de paquets perdus sur la canal de contrôle et le RTT des paquets ARP. Sur la Figure 12, on voit comment la solution proxy-ARP est pénalisée quand le taux de pertes sur le plan de contrôle augmente. La solution proxy-ARP souffre du fait que la connexion entre le commutateur OF et le contrôleur soit si sensible aux pertes de paquets, ralentissant sensiblement le débit entre eux, et par conséquent augmente le temps de réponse aux requêtes ARP.

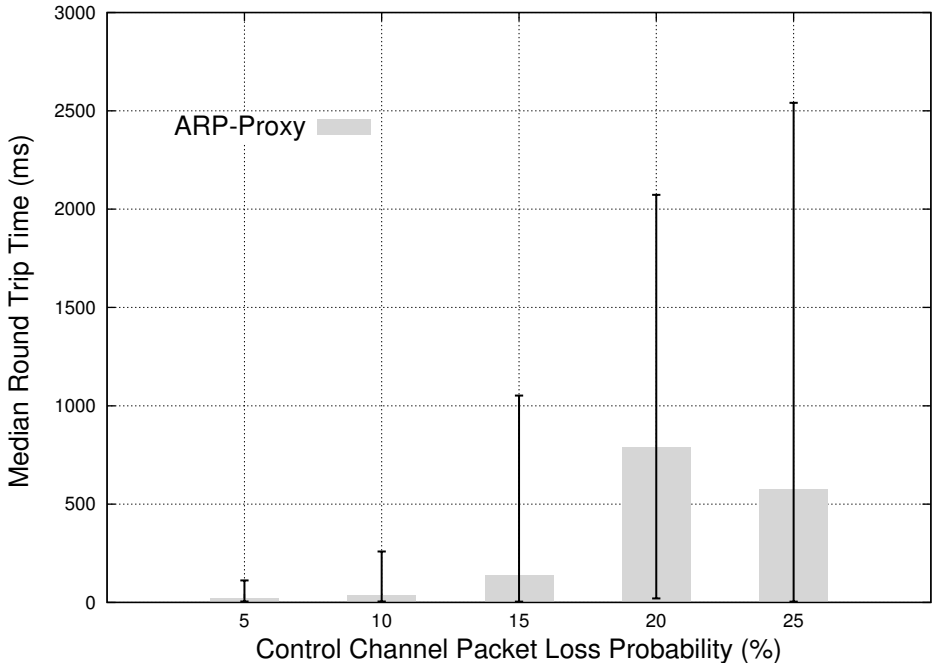

Figure 12. RTT d'un paquet ARP quand un proxy-ARP est actif, le cache ARP est vide, et le canal de contrôle subit différent taux de pertes de paquets.

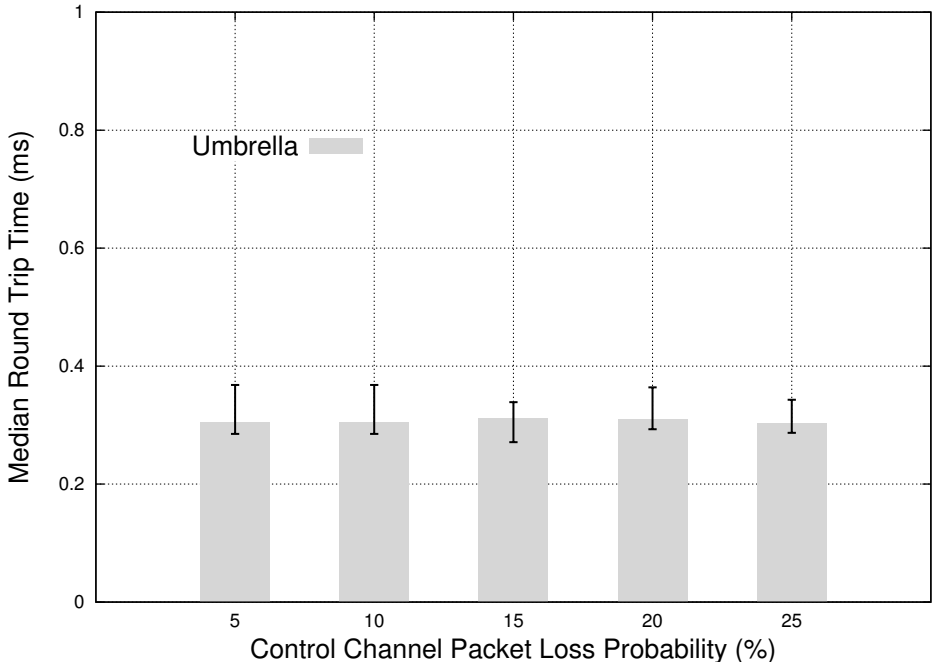

Figure 13. RTT d'un paquet ARP quand Umbrella est utilisée, le cache ARP est vide, et le canal de contrôle subit différent taux de pertes de paquets.

3) Impact du plan de contrôle sur le débit du plan de données *:* Les tests précédents ont démontré la différence fondamentale qui existe entre une approche basée sur un proxy-ARP (utilisée dans de nombreuses approches SDN) et Umbrella. Cependant, en pratique, le

comportement du canal de contrôle (en particulier en terme de délai) est plus complexe que ce qui a été étudié dans les deux cas précédents, et l'impact sur le plan de données n'est pas aussi prédictible. Pour examiner ces différences sur un scénario plus réaliste, nous évaluons maintenant l'impact potentiel du plan de contrôle sur les débits au niveau du plan de données dans une fabrique d'IXP.

Comme pour le scénario représenté sur la Figure 1, nous avons instancié sur Mininext 100 routeurs pairs et les avons connectés à l'Open vSwitch. Chaque routeur a été connecté à un hôte virtuel différent, et nous avons configuré la table de flux du commutateur de manière à garantir une connectivité totale au niveau du plan de données, ainsi qu'une règle par défaut pour rediriger le trafic ARP vers le contrôleur Ryu sur lequel s'exécute le proxy-ARP. La bande passante de chaque lien a été fixée à 10Mbps pour pouvoir générer suffisamment de trafic sur le plan de données et observer ses éventuelles dégradations. Chaque hôte ouvre également une session Iperf vers les autres routeurs. Les débits agrégés des flux sont ensuite mesurés au niveau du commutateur.

Cette expérimentation a été réalisée pour deux cas différents : un canal de contrôle idéal (sans perte ni délai) et un canal de contrôle introduisant 50ms de délai. La Figure 14 illustre la réduction des débits au niveau du plan de données due aux délais du plan de contrôle. Cette baisse des débits est due à l'expiration des entrées des caches ARP, et conduit à des perturbations pour certains hôtes. Ceci illustre une nouvelle fois combien une séparation forte entre les plans de contrôle et de données peut être bénéfique et peut prévenir de nombreux problèmes du plan de données.

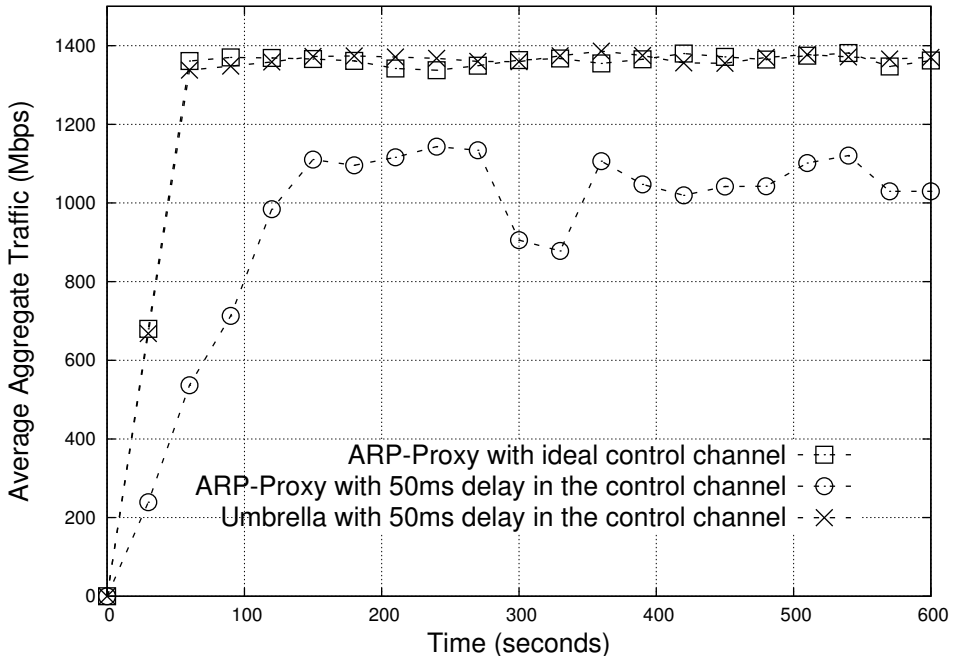

Figure. 14. Impact relatif des trafics des plans de données et de contrôle

6. Déploiement et évaluation d'Umbrella sur TouIX : de TouIX à TouSIX

Le réseau TouIX était une infrastructure interconnectée autour de 4 POPs (Point of Presence) et interconnectée avec deux autres IXP : FranceIX et LyonIX. TouIX a été renommé TouSIX après sa migration vers une approche OF. Depuis mai 2015, TouSIX est le premier IXP européen (et à ce jour le seul) à exploiter la technologie OF pour ses opération quotidiennes. L'architecture TouSIX utilise l'approche Umbrella. Cette partie commence par présenter les travaux de migration de TouIX vers une solution OF, puis montre ensuite comment TouSIX et son approche Umbrella fonctionnent et résolvent les problèmes présents avec l'ancien réseau TouIX.

6.1. TouIX

La partie supérieure de la Figure 15 montre la topologie de TouIX. Le site primaire a été Cogent où le BIRD RS a été installé. Les équipement CISCO[7] utilisés dans la fabrique ont été configurés avec trois VLAN : un VLAN admin, un VLAN TouIX et un VLAN IX pour les 2 IXP externes connectés. Comme les équipements CISCO installés étaient vieillissants et plus réellement appropriés pour la fabrique, il a été décidé de refondre complètement le réseau et de créer un tout nouvel IX pour faciliter sa gestion

En particulier, les mauvaises configurations de membres (i.e., avec des boucles non souhaitées) affectaient l'ensemble des opérations de la fabrique, et rendaient délicate la gestion de l'IXP. Plusieurs « tempêtes ARP » (ARP storm) ont été subies. L'absence d'un système de monitoring moderne et efficace couplée à des apparitions irrégulières de ce phénomène l'ont rendu difficile à détecter. Les commutateurs CISCO ne disposaient pas des règles de filtrage requises pour empêcher ce trafic non voulu de perturber la fabrique. Protéger la fabrique du trafic indésirable est une nécessité, non seulement pour protéger les routeurs pairs, mais aussi pour améliorer la stabilité et la facilité de gestion de toute l'infrastructure. Umbrella a été notre réponse pour solutionner ces problèmes.

[7] TouIX used Cisco WS-C3550-12G (TLS00), WS-C3560G (Cogent), WS- C2960G-24TC-L (Zayo) and WS-C2960G-24TC-L (Hotel Telcom).

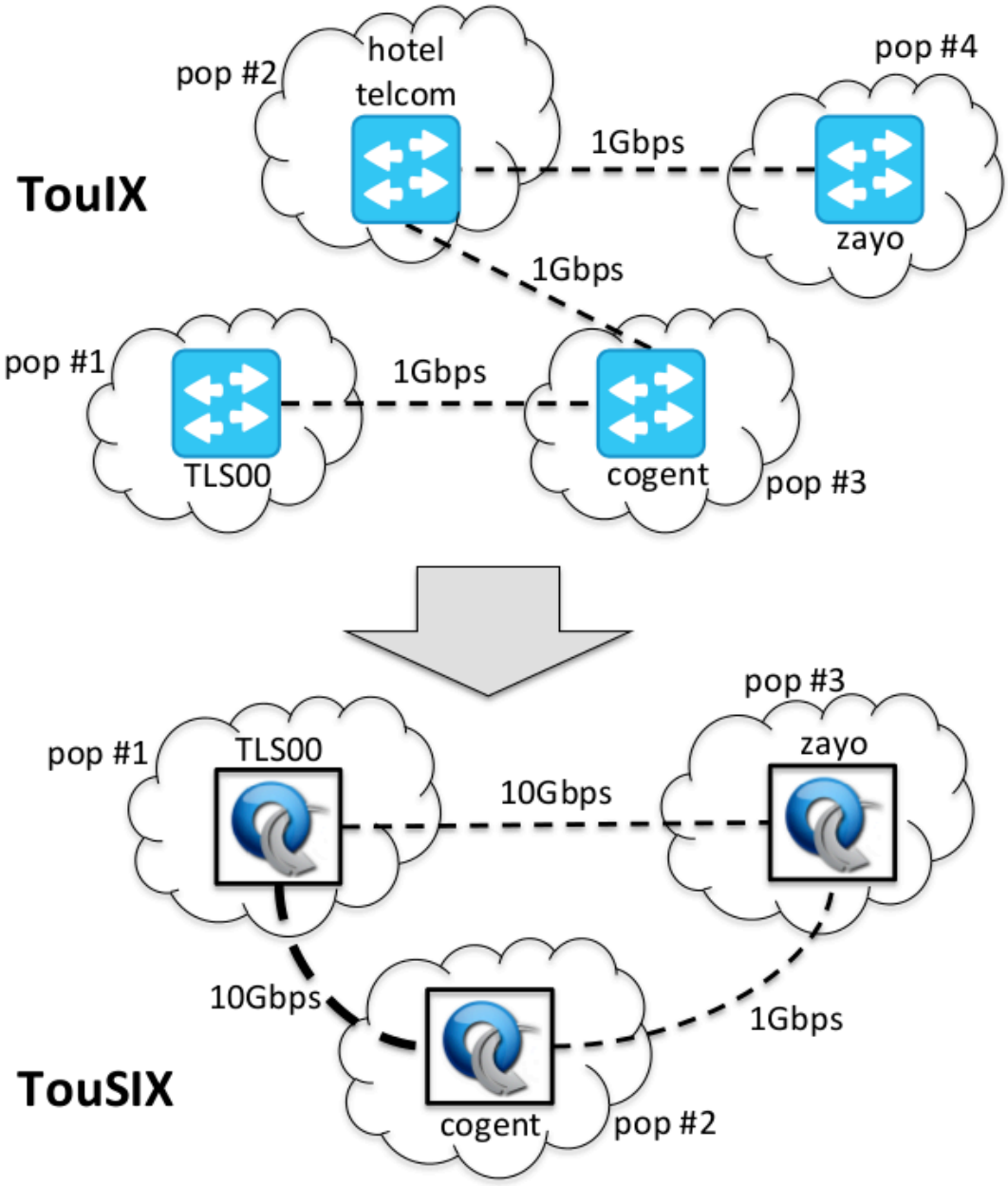

Figure 15. D'un IXP classique à un IXP complètement opéré avec OF

6.2. TouSIX : un IXP totalement opéré par SDN

La nouvelle topologie de TouSIX a trois PoPS, avec un commutateur OF sur chacun d'eux qui agit en tant que commutateurs de bordure. Il n'y a pas de routeur de cœur (cf. partie basse de la Figure 15). Les trois commutateurs ont été programmés selon les principes d'umbrella et remplacent les quatre anciens commutateurs de TouIX. Le site de l'*Hotel des Télécoms* a été supprimé durant cette migration et son unique membre a été connecté directement au commutateur de *Cogent*.

Une longue période de test était requise pour valider cette nouvelle configuration système et sa stabilité avant de l'utiliser en production.

Au début de la campagne de test, seuls les commutateurs Pica8 intégraient de façon matérielle un agent Open vSwitch, faisant du commutateur Pica8 P-3290 le choix naturel pour TouSIX. Il a été décidé de faire reposer TouSIX sur le commutateur logiciel le plus largement utilisé dans le monde, de plus avec un niveau de performance similaire à celui de solutions hardware. Toutefois, la dernière version du système PicOS au début de la campagne de test ne permettait pas d'installer des règles mentionnant le champ ipv6 nd. Cette option est requise pour mettre en œuvre les mécanismes d'Umbrella sur le trafic IPv6 (cf. section 5.1). De plus, le système logiciel PicOS vidait entièrement la table de flux lors de reboots matériels, ou lorsque la connexion avec le contrôleur était perdue. En collaboration avec Pica8, ces problèmes ont pu être résolus, et ont conduit à une nouvelle version de PicOS, i.e., 2.6.

La période de tests a duré quelques mois, et la version de la nouvelle fabrique a pu être installée. Les commutateurs Pica8 fonctionnant en mode Open vSwitch ont été placés au dessus des commutateurs Cisco. Les commutateurs Pica8 servent pour le plan de données, et les Cisco ont été conservés pour transporter le trafic du plan de contrôle. Grâce aux nouveaux commutateurs, la bande passante a été augmentée d'un facteur 2 ou 3 entre les commutateurs de bordure. De plus, la connexion entre *TLS00* et *Cogent* a été améliorée avec le mécanisme d'agrégation de liens (Link AGgregation, LAG). Les câbles des membres ont été migrés des commutateurs Cisco vers les Pica8, avec seulement quelques secondes d'interruption de service pour chaque membre. Les commutateurs Pica8 ont été configurés pour ne pas vider la table des flux lorsque l'agent OF perd sa connexion avec le contrôleur OF. Le contrôleur n'est cependant plus le point de faiblesse du réseau grâce à la forte séparation entre les plans de contrôle et de données. Le déploiement en cours repose sur Ryu [18], le contrôleur open-source des NTT Labs, alors qu'on attend de pouvoir évaluer le contrôleur ONOS de ON.Labs [30] et son parallélisme. D'autre part, les développeurs utilisent l'API REST Pica8 pour communiquer avec une interface graphique (GUI : Graphical User Interface), qui simplifie beaucoup les tâches opérationnelles classiques.

1) Expérience avec un déploiement réel : La fabrique Umbrella de TouSIX a fonctionné sans problème et avec un support réduit des administrateurs. Le Tableau 2 trace certains résultats avec Umbrella en fonctionnement réel : conversion du trafic ARP broadcast en unicast, blocage du trafic non désirable, annonce de 399 préfixes IP. Umbrella réduit ainsi le volume total moyen du trafic ARP qui transite sur la fabrique de 97%. De plus, une application de gestion – le TouSIX-manager – a été développée pour permettre aux membres de configurer leur peering directement, en générant automatiquement les règles OF pour le commutateur SDN, ainsi que pour configurer le RS.

Le Tableau 3 montre le trafic moyen et crête sur TouSIX. TouSIX concentre le trafic de 14 nœuds, comprenant les routeurs et les serveurs (même s'ils ne sont pas tous actifs à un instant donné, toutes les règles correspondantes restent installées). Les commutateurs doivent intégrer des règles pour le forwarding MAC, le routage ARP et le routage ICMPv6 NS. Comme seul IPv4 était en production à cette époque, le nombre total de règles par commutateurs est de 28. Umbrella permet aussi de limiter la consommation de ressources qui présente des niveaux d'utilisation CPU sans pics, et stable aux alentours de 8%. A l'inverse, les commutateurs du TouIX subissaient des pics de charge atteignant les 100% de leur capacité. Ces pics de charge CPU des anciens commutateurs TouIX étaient le résultat de problèmes de convergence de différents protocoles du plan de contrôle.

En terme de temps de récupération sur erreur, lorsqu'un commutateur avec une table vide se

connecte au contrôleur, installer les règles de flux (i.e., l'appel à la base de données et d'émission des règles vers le commutateur) prend environ 5.6 secondes, et un reboot hard prend environ 64 secondes. Avec Umbrella, la connexion d'un nouveau routeur à TouSIX marche selon le principe "plug & play" : l'administrateur installe automatiquement les règles de flux pour tout router « approuvé », car le contrôleur connaît déjà sa configuration. Par conséquent, le temps entre le moment où un commutateur est connecté, et son passage en mode opérationnel est négligeable. A l'inverse, si un routeur est connecté sans l'approbation de l'administrateur, tout son trafic est éliminé. Comme le TouSIX-Manager opère la topologie avec des procédures proactives, si le lien est perdu au niveau du canal de contrôle, toutes les règles déjà en place sur le commutateur demeurent. Nous avons testé l'effet de l'arrêt du TouSIX-manager et avons vérifié que les statistiques sur le trafic restent les mêmes sans avoir d'impact sur les opérations de la fabrique.

	TouIX	TouSIX
Max. (Pkt/s)	14,96	3
Average (Pkt/s)	8,51	1,18
Min. (Pkt/s)	1,1	0

Tableau 2. Trafic ARP (en paquets par seconde) dans TouIX et dans TouSIX.

	Commutateur 1	Commutateur 2	Commutateur 3
Peak (B/s)	4874601415	5367083113	268646
Average (B/s)	1975571	43955853	103

Tableau3. Trafic moyen et crête (en octets par seconde) dans chacun des commutateurs TouSIX.

7. Etat de l'art

Introduire OF dans le monde des IXP est une idée très récente. Gupta et al. [2], [3] ont développé un point d'échange utilisant SDN (SDX) pour permettre la mise en place de règles plus évoluées que les règles de forwarding hop-by-hop conventionnelles. Cette solution a montré qu'il est possible d'implémenter des polices représentatives pour des centaines de participants tout en garantissant des temps de réponse à des changements de configuration et des modifications de routage inférieurs à la seconde. Pour y parvenir, la version multi-tables du prototype iSDX considère un scénario dans lequel tous les participants sont connectés à un unique commutateur. En réalité, cependant, il peut y avoir des sauts multiples dans l'infrastructure de l'IXP. Umbrella, avec son approche par commutation de labels, peut servir de support fiable de l'architecture iSDX en forwardant directement les paquets de découverte de la localisation à tous les participants, ce qui rend Umbrella et iSDX complémentaires. En

particulier, après qu'iSDX ait pris sa décision quant au port de sortie, l'adresse MAC de destination peut être ré-écrite en utilisant les mécanismes d'umbrella. Les paquets seront alors délivrés en suivant le chemin encodé dans le champ d'adresse MAC destination. Cependant, l'intégration proposée nécessiterait qu'un commutateur OF supplémentaire s'occupe de des requêtes ARP pour la recherche des prochains sauts virtuels et qui vont être pris en charge par un proxy-ARP. Comme Umbrella transforme les paquets broadcast ARP en unicast, les requêtes ARP doivent avoir le chemin vers le proxy-ARP encodé au niveau de l'adresse MAC de destination. Cela améliorerait la flexibilité, car le proxy-ARP n'a pas besoin d'être connecté à un commutateur spécifique de la fabrique de l'IXP.[8]

Le projet cardigan [1] a implémenté en hardware une politique de déni par défaut permettant de filtrer à partir d'une vérification RPKI les routes annoncées par des équipements connectés à la fabrique. Alors que cette approche offre le niveau de protection requis pour une fabrique d'IXP stable, il est moins fiable pour les IXPs qui voudraient rester neutres par rapport aux principes de forwarding du trafic.

L'architecture B4 de Google [31] essaie aussi de régler les problèmes de scalabilité d'OF en forwardant es fuxs de façon proactive sur le modèle d'Umbrella. Toutefois, l'architecture B4 requiert que le contrôleur traite les décisions de façon active, alors qu'Umbrella est une solution pour les IXPs dont la fabrique ne prend pas part au processus de décision de peering.

Le routage au niveau MAC dans les réseaux OF est un domaine de recherche nouveau. Schwabe et al. [32] ont montré que l'adresse MAC de destination peut être utilisée comme un label universel dans les environnements SDN, et les caches ARP des hôtes peuvent être utilisés comme une table de labels entrants, réduisant ainsi les tables de forwarding des équipements réseaux. Agarwal et al. [33] ont démontré qu'en utilisant les adresses MAC comme des labels de forwarding opaques, un contrôleur SDN peut utiliser les grandes tables de forwarding MAC pour gérer pléthores de chemins grain fin. Même si ces approches présentent des propriétés intéressantes pour les réseaux à grande échelle, elles reposent sur un mécanisme de proxy-ARP, ce qui implique les limitations déjà discutées dans ce rapport.

8. Conclusion

Umbrella est une solution qui améliore la scalabilité, la fiabilité et simplifie le management des IXPs. En gérant une partie du trafic de contrôle directement dans le plan de données, notre approche complète les architectures d'IXP existantes et illustre les avantages d'une séparation plus forte entre les plans de contrôle et de données pour la gestion des IXPs. Nous avons également montré l'applicabilité d'Umbrella au travers d'un déploiement convaincant et fructueux sur notre réseau TouSIX. L'architecture Umbrella est scalable et peut être utilisée de manière totalement neutre pour les flux de niveau 3 et au dessus, ce qui est un élément important des IXPs aujourd'hui et vraisemblablement dans le futur.

La suite de ce projet vise à améliorer encore la scalabilité de la solution Umbrella. Mais elle

[8] La fonction de proxy-ARP dans iSDX est essentielle pour compresser les flux en classes d'équivalence de forwarding.

ambitionne aussi d'améliorer la robustesse et la sécurité de l'IXP et notamment du contrôleur. En effet, même si Umbrella continue à fonctionner dans le cas où le canal de contrôle n'assure plus la connexion entre le contrôleur et les commutateurs OF du plan de données, ou si le contrôleur est inactif ou en panne, il n'est pas possible de changer la configuration et les règles des commutateurs OF. Nous voulons donc rendre le contrôleur plus fiable et plus sécurisé. Pour cela, nous venons de lancer une étude dans laquelle nous imaginons différentes techniques de distribution du contrôleur, ce qui permettra donc (1) de pouvoir soutenir des taux de trafic de contrôle bien supérieur à ce qu'ils sont aujourd'hui, pouvoir soutenir des phases d'attaques d'une ou plusieurs composante du contrôleur, et ce sans jamais arrêter aucun des services de l'IXP que ce soit sur les plans de contrôle ou de données. De nouvelles techniques de détection d'attaques seront également imaginées et évaluées. L'objectif est de projeter Umbrella et TouSIX vers le futur, en essayant de disséminer notre architecture et nos outils (notamment le TouSIX-manager) le plus largement possible. La nature open-source de TouSIX-manager a été choisie en ce sens.

D'autre part, le projet TouSIX entretient une collaboration étroite avec le projet européen H2020 ENDEAVOUR [34]. ENDEAVOUR envisage de construire une nouvelle fabrique dans laquelle iSDX est utilisé au niveau du processus de décision des participants, et Umbrella pour fournir un mécanisme de forwarding fiable et scalable dans la fabrique de l'IXP. En effet, Umbrella propose une architecture pour réseaux SDN qui est moins dépendante du plan de contrôle, et agit comme un superviseur intelligent plutôt que comme un point de décision critique. Cette combinaison des deux architectures iSDX et Umbrella est en cours de réalisation dans le cadre du projet H2020 ENDEAVOUR.

9. Références

[1] J. Stringer, D. Pemberton, Q. Fu, C. Lorier, R. Nelson, J. Bailey, C. Correa, and C. Esteve Rothemberg, "Cardigan: SDN distributed routing fabric going live at an Internet exchange," in *ISCC*. IEEE, 2014.

[2] A. Gupta, L. Vanbever, M. Hahbaz, S. Donovan, B. Schlinker, N. Feam- ster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A Software Defined Internet Exchange," in *SIGCOMM*. ACM, 2014.

[3] A. Gupta, R. MacDavid, R. Birkner, M. Canini, N. Feamster, J. Rexford, and L. Vanbever, "iSDX: An Industrial-Scale Software Defined Internet Exchange Point," in *NSDI*. USENIX, 2016.

[4] B.Ager,N.Chatzis,A.Feldmann,N.Sarrar,S.Uhlig,andW.Willinger, "Anatomy of a Large European IXP," in *SIGCOMM*. ACM, 2012.

[5] N. Chatzis, G. Smaragdakis, J. Bo ̈ttger, T. Krenc, A. Feldmann, and W. Willinger, "On the Benefits of Using a Large IXP as an Internet Vantage Point," in *IMC*. ACM, 2013.

[6] H. D. Vu and J. But, "How rtt between the control and data plane on a sdn network impacts on the perceived performance," in *International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE Computer Society, 2015.

[7] K. He, J. Khalid, A. Gember-Jacobson, S. Das, C. Prakash, A. Akella, L. E. Li, and M. Thottan, “Measuring control plane latency in sdn- enabled switches,” in *Symposium on Software Defined Networking Research (SOSR)*. ACM, 2015.

[8] B.Ager,N.Chatzis,A.Feldmann,N.Sarrar,S.Uhlig,andW.Willinger, “Anatomy of a Large European IXP,” in *SIGCOMM*. ACM, 2012.

[9] N. Hilliard, E. Jasinska, R. Raszuk, and N. Bakker, “Internet Exchange Route Server Operations,” Internet- Draft, Tech. Rep. [Online]. Available: https://tools.ietf.org/html/ draft- ietf- grow- ix- bgp- route- server- operations- 03

[10] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger, “Peering at Peerings: On the Role of IXP Route Servers,” in *IMC*. ACM, 2014.

[11] M. Wessel and N. Sijm, “Effects of IPv4 and IPv6 address resolution on AMS-IX and the ARP Sponge,” Master’s thesis, Universiteit van Amsterdam, the Netherlands, 2009.

[12] “FranceIX website,” https://www.franceix.net/en/events- and- news/news/ franceix-outage- notification/, [Online; accessed 29-Jan-2016].

[13] J. C. Cardona and R. Stanojevic, “IXP Traffic: A Macroscopic View,” in *LANC*, 2012.

[14] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” *CCR*, 2008.

[15] V. Boteanu and H. Bagheri, “Minimizing ARP traffic in the AMS-IX switching platform using OpenFlow,” Master’s thesis, Universiteit van Amsterdam, the Netherlands, 2013.

[16] I. Pepelnjak, “Could IXPs Use OpenFlow to Scale?” MENOG, 2012.

[17] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster, E. Katz-Bassett, and M. Yu, “Try Before you Buy: SDN Emulation with (Real) Interdomain Routing,” in *ONS*. USENIX, 2014. [Online]. Available: https://www. usenix.org/conference/ons2014/technical-sessions/presentation/schlinker

[18] “Ryu SDN controller,” http://osrg.github.io/ryu, [Online; accessed 29- Jan-2016].

[19] AMS-IX, “Allowed Traffic Types on Unicast Peering LANs,” http://ams-ix.net/technical/specifications-descriptions/allowed-traffic, [Online; accessed 29-Jan-2016].

[20] M. Hughes, M. Pels, and H. Michl, “Internet Exchange Point Wishlist, ”https://www.euro- ix.net/ixps/ixp- wishlist/, 2015, [Online; accessed 29- Jan-2016].

[21] Open-IX, “IXP Technical Requirements OIX-1,” http://www.open-ix.org/standards/ixp-technical- requirements, [Online; accessed 29-Jan-2016].

[22] “OpenFlow Switch Specification,” http://www.openflow.org/documents/openflow-spec- v1.0.0.pdf, [Online; accessed 29-Jan-2016].

[23] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6),” Tech. Rep., 2007. [Online]. Available: http://tools.ietf.org/html/rfc4861

[24] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, “Enabling Fast Failure Recovery in OpenFlow Networks,” in *DRCN*. IEEE, pp. 164–171.

[25] N. L. V. Adrichem, B. J. V. Asten, and F. a. Kuipers, “Fast Recovery in Software-Defined Networks,” *EWSDN*, pp. 61– 66, 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/ wrapper.htm?arnumber=6984053

[26] “PeeringDB website,” http://www.peeringdb.com, [Online; accessed 29- Jan-2016].

[27] A. Lodhi, N. Larson, A. Dhamdhere, C. Dovrolis, and k. Claffy, “Using peeringDB to Understand the Peering Ecosystem,” *CCR*, 2014.

[28] B. Augustin, B. Krishnamurthy, and W. Willinger, “IXPs: Mapped?” in *SIGCOMM*. ACM, 2009.

[29] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger, “There Is More to IXPs than Meets the Eye,” *CCR*, vol. 43, no. 5, pp. 19–28, 2013.

[30] “ONOS official website,” http://onosproject.org, [Online; accessed 29- Jan-2016].

[31] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Ho ̈lzle, S. Stuart, and A. Vahdat, “B4: Experience with a Globally-deployed Software Defined Wan,” in *SIGCOMM*. ACM, 2013.

[32] A.SchwabeandK.Holger, “UsingMACAddressesAsEfficientRouting Labels in Data Centers,” in *HotSDN*. ACM, 2014.

[33] K. Agarwal, C. Dixon, E. Rozner, and J. Carter, “Shadow MACs: Scalable Label-switching for Commodity Ethernet,” in *HotSDN*. ACM, 2014.

[34] ENDEAVOUR official website, « http:// », [Online ; accessed 7-July-2017]