TRAVAIL DE BACHELOR 2011

Prototype dual-stack IPV4/6 sur un backbone MPLS-VPN (réseau)

Rapport final

Julien Tissot

29/07/2011

Professeur responsable :	Dr. Stephan Robert, HEIG-VD, Yverdon-les-Bains
Travail suivit par :	Jérôme Vernez, SIEN, Neuchâtel

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences
Western Switzerland

Table des matières

1 Résumé

Ce travail, effectué pour le SIEN (Service Informatique de l'Entité Neuchâteloise), a comme but de présenter le nouveau protocole IPv6 et de proposer des solutions pour une migration douce. L'idée est de créer des prototypes calqués sur le réseau du SIEN pour découvrir IPv6, puis pour chercher comment la transition pourra être effectuée. Ce travail présentera en particulier les différences avec IPv4 sur le plan théorique dans un premier temps puis sur le plan la pratique. On verra donc comment sont construites les adresses, les plans d'adressage ainsi que les différentes transitions possibles telles que le dual-stack, technique conseillée pour une transition complète du réseau.

Grâce aux prototypes, on découvrira et analysera ICMPv6 qui remplace ARP. Avec ses nouveaux types, il permet de récupérer plusieurs informations importantes des routeurs. Il permet aussi d'alléger les serveurs DHCP en proposant différentes possibilités de récupération d'adresses IP. Comme les hôtes doivent se construire des adresses par eux-mêmes, il suffit aux routeurs de leur donner le préfix du réseau. Il est ainsi possible de se passer des serveurs DHCP. Si l'envoi d'options aux hôtes est nécessaire, toujours grâce à ICMPv6, les routeurs peuvent indiquer à ces derniers qu'ils les cherchent auprès du serveur. Dans les cas où cela s'impose, le mode DHCP tel qu'il est connu avec IPv4 est aussi possible. Cette nouvelle flexibilité est une grande force d'IPv6.

Le travail se voulant aussi pratique, on cherchera ce qu'il est possible de faire sur du matériel Cisco et quelles sont les limitations. Etant donné que le réseau du SIEN est composé de plusieurs VRF (Virtual Routing and Forwarding), on a vite remarqué qu'il n'est pas encore possible de créer un réseau similaire à celui du SIEN. La principale limitation se trouve dans le fait qu'OSPFv3, la version qui intègre IPv6, ne supporte pas les VRF. De plus, tous les équipements (en particulier les switchs L2-L3) ne supportent pas les VRF IPv6. La solution, pour créer un réseau multi-VRF, est donc de faire du routage statique sur les appareils qui l'intègre, ce qui a été fait.

Le backbone BGP/MPLS est aussi analysé, car il constitue le centre du réseau et est donc très critique. S'il n'est pas possible qu'il soit dual-stack dû au fait que MPLS ne sait pas utiliser IPv6 pour l'échange des tags, il fonctionne tout de même totalement grâce à la technique d'intégration 6VPE qui permet d'envoyer les routes de plusieurs VRF IPv6, en utilisant IPv4, à des voisins dual-stack. Cela permet de ne pas devoir modifier les configurations des routeurs internes au backbone. Ces derniers n'ont aucune idée du protocole utilisé vu qu'ils s'appuient uniquement sur les tags pour transmettre les paquets.

Finalement, le multicast est aussi abordé. Peu d'appareils le supportent et n'ayant pas d'application pouvant l'utiliser, l'analyse se porte principalement sur les différences avec IPv4 et les compatibilités en fonction des équipements. Là aussi, tout n'est pas encore prêt du côté de chez Cisco.

L'analyse des performances effectuées montre que, sur tous les équipements, le trafic IPv6 n'est pas plus gourmand en ressources que celui d'IPv4. Mais cela peut être faussé par le peu trafic qu'il a été possible générer. Par contre, le dual-stack utilise logiquement plus de mémoire étant donné qu'il doit intégrer deux fois plus de tables de routages.

En conclusion, la solution de migration est proposée. Dû au fait que le backbone est prêt à recevoir IPv6 et que le SIEN ne se trouve pas dans l'urgence, il est proposé d'utiliser la technique « edge-to-core » c'est-à-dire de commencer par le centre puis d'aller en fonction des capacités et des besoins vers les extrémités.

2 Introduction

Ce travail de Bachelor est fait dans le but d'aider le SIEN (Service Informatique de l'Entité Neuchâteloise) à se préparer à l'arrivée d'un nouveau protocole d'adressage Internet. Ce protocole, dont on entend souvent parler ces derniers temps, s'appelle IPv6. Etant une avancée très grande dans les télécommunications réseau, il révolutionne la base d'une architecture qui est en place depuis passé 30 ans à présent. Comme lors de toute révolution, certaines personnes y sont réticentes alors que d'autres ont envie de découvrir, voire d'en être les précurseurs et c'est dans ces derniers que se trouve le SIEN par rapport à ses homologues. Il va donc nous falloir découvrir comment cela fonctionne, qu'est-ce que l'on peut faire avec et s'il va être bientôt possible de l'implémenter sur les réseaux de production. On va donc essayer de l'apprivoiser le mieux possible pour voir ce que cette révolution peut apporter et comment on va pouvoir l'intégrer en douceur. Pour ce faire, nous allons analyser ses bases et ses principales modifications par rapport à son prédécesseur. Ensuite, il va falloir observer comment réagissent tous les composants qui touchent directement à ce nouveau protocole. On pense principalement au serveur dédié à l'adressage ou encore aux différents protocoles de routage. Ce rapport sera orienté sur les installations réseaux alors que mon collègue Steve Lienhard se penchera lui sur la partie services et sécurité. Etant donné que nous avons travaillé conjointement, il est fort possible que nos rapports se ressemblent sur certains points.

Pendant le travail, afin d'avoir quelques avis extérieurs, nous sommes allés à un séminaire des SIG (Services Industriels de Genève) sur IPv6 lors duquel la spécialiste, Silvia Hagen, a donné son point de vue très intéressant. On retrouvera donc dans ce travail certaines remarques de sa part.

3 IPv6, les bases

3.1 IPv6, pourquoi ?

Tout appareil voulant être connecté à Internet se doit d'avoir une adresse IP (Internet Protocol) afin de pouvoir être joint un peu comme une personne et son adresse postale. Aujourd'hui, les IP sont codées sur 32 bits ce qui en donne un nombre théorique maximal de 2^{32}ou $4.3 * 10^9$ ou 4.3 milliards. Sachant que, de nos jours, on désire y connecter toutes sortes d'appareils autres que des ordinateurs comme les imprimantes, les téléphones portables, les systèmes d'alarme, et même gentiment les voitures, les cuisinières et bien d'autres encore, le nombre d'adresses libres a déjà atteint zéro depuis février dans les plus grandes instances de distribution. Mais pour l'instant, tous les étages inférieurs de la hiérarchie de distribution ont encore quelques petites réserves. Même s'il existe des techniques pour n'avoir besoin que d'une adresse pour un groupe d'appareils (Network Address Translation) et ainsi économiser celles que l'on appelle publiques, l'épuisement des adresses se fait de plus en plus ressentir en fonction des régions du globe et c'est donc pourquoi l'Internet Engineering Task Force (IETF) s'est penché sur le problème depuis le début des années 1990. En 1995, les principales spécifications de la nouvelle version d'IP (IPv6) sont déjà publiées et sont régulièrement mises-à-jour depuis. Aujourd'hui, la plupart des entreprises commencent à se pencher sérieusement sur le problème et les plus grosses (Google, Yahoo) et certaines plus petites (qoqa.ch) sont déjà connectées et atteignables en IPv6.

En plus d'agrandir la plage d'adresses, IPv6 propose un grand nombre d'améliorations et de nouveaux services par rapport à son prédécesseur comme la simplification de l'entête IP, l'auto-configuration, la mobilité ou la sécurité intégrée. Toutes ces options seront très utiles pour la gestion des réseaux.

3.2 Les adresses

Contrairement aux adresses IPv4 qui sont donc codées sur 32 bits, IPv6 est parti sur une base beaucoup plus grande avec 128 bits ce qui donne $3.4 * 10^{38}$ possibilités (contre $4.3 * 10^9$ avant). La notation d'une adresse se fait de la façon suivante en IPv4 : 4 groupes de nombres décimaux entre 0 et 256 (10.42.132.200). Celle d'IPv6 a été modifiée, on a le droit à 8 groupes de 4 nombres hexadécimaux séparés par des double-points (2001:3d44:0c8a:0000:0000:1fff:5b83:0a02). Pour simplifier l'écriture et la lecture, il faut enlever les 0 en début de groupe et remplacer, au maximum une fois, les groupes de 0 consécutifs par '::'. L'adresse présentée avant nous donne donc : 2001:3d44:c8a::1fff:5b83:a02.

Il est sûr que le nombre total d'adresses possible avec IPv6 et calculé précédemment n'est que théorique et ne pourra pas contenir autant d'hôtes pour plusieurs raisons. Comme on le verra, tous les sous-réseaux ont des masques de 64 bits ce qui fait un nombre extrêmement élevé d'adresses d'hôtes qui ne seront jamais utilisées. Donc, même si ces adresses existent, on ne mettra jamais 2^{64} hôtes dans un même sous-réseau. De plus, chaque carte réseau (voulant pouvoir se connecter sur d'autres sous-réseaux que le sien) aura au moins deux adresses comme on le verra ci-dessous.

Contrairement à IPv4, les cartes réseaux peuvent avoir plusieurs adresses provenant de différentes ou de la même classe : les adresses de liaison locale, les adresses de site et les adresses globales. Nous allons voir en détail ces différentes classes mais avant cela, il faut encore noter que le

broadcast n'existe pas en IPv6 par contre le multicast a été grandement amélioré. Toutes ces différentes adresses sont principalement définies dans la RFC 4291.

3.2.1 Les adresses de liaisons-locales

Toutes les cartes réseaux avec IPv6 activé, même sans qu'un média y soit branché, possèdent déjà une adresse IPv6 appelée « adresse de liaison-locale ». Cette adresse va permettre à l'interface de communiquer avec toutes les autres interfaces qui sont dans le même domaine de diffusion (ce sont les routeurs qui ne retransmettent pas les paquets ayant ces adresses comme source) et c'est elle qui est utilisée au moment de la connexion d'un média pour découvrir (entre autres) les routeurs comme on le voit sur la Figure 1 grâce au « Neighbor Discovery Protocol » (cf. ICMPv6).

```
0.307376   ::                          ff02::1:fffc:4c9c     ICMPv6 Neighbor solicitation for fe80::ac90:758c:83fc:4c9c
0.307410   fe80::ac90:758c:83fc:4c9c   ff02::2               ICMPv6 Router solicitation from 00:25:4b:a9:ba:04
```

Figure 1 : Premiers paquets envoyés par une interface qui se connecte à un réseau IPv6

Le premier paquet, « Neighbor solicitation », sert à s'assurer que notre adresse de liaison locale est bien unique dans le domaine de diffusion car si elle ne l'est pas, il faudra en construire une autre. Ensuite, l'interface utilise cette adresse pour faire un « Router solicitation », découverte de routeur. Ces adresses commencent par « fe80::/10 » et on utilise normalement uniquement les 64 derniers bits qui sont, soit construits aléatoirement, soit en fonction de l'adresse MAC comme l'explique M. Lienhard dans son travail.

3.2.2 Les adresses de site

Les adresses de site en IPv6 peuvent être comparées aux adresses privées en IPv4 mais au vu du nombre d'adresses globales (publiques) disponibles et après des années de tests, cette classe d'adresse a récemment été déconseillée d'utilisation. On pourrait tout de même les utiliser si l'on veut un adressage local pour des entreprises se trouvant dans différents pays, voir même continents. Cela impliquerait aussi que chaque interface ait, au minimum, trois adresses (liaison locale, adresse de site et adresse globale). On peut les reconnaitre à leurs adresses qui commencent par fec0::/10. Nous n'allons pas nous étendre sur cette classe étant donné qu'elle ne sera pas utilisée.

3.2.3 Les adresses globales

Les adresses globales sont routées sur Internet et ce sont aussi celles que l'on va utiliser dans notre réseau pour atteindre les différents hôtes. Elles commencent par « 2000::/3 ». La théorie de base nous dit qu'elles sont construites de la manière suivante : Le masque de sous-réseau est de 64 bits. Ces derniers sont assignés aux particuliers par leur fournisseur d'accès Internet. Ceux qui le désirent et les entreprises se verront assigner un (ou des) réseau avec un masque de 48 ce qui leur permettrai déjà de faire 65'536 sous-réseaux. Un étage plus haut, les fournisseurs d'accès recevront eux un réseau /32 qu'ils pourront diviser en 65'536 réseaux clients. Finalement les RIR (Regional Internet Registry) recevront des blocs avec des masques variant entre /12 et /23. Ces tailles de masque sont celles de base, mais les gros fournisseurs d'accès pourront recevoir plusieurs blocs /32 consécutifs alors que les petits pourraient ne recevoir que des /48. Ensuite, il y a des cas spéciaux : Par exemple, certaines institutions recevront des blocs indépendants des fournisseurs comme c'est le cas avec la confédération qui distribue ensuite de plus petits blocs aux cantons. Cette hiérarchisation des adresses va permettre de simplifier significativement les tables de routages en utilisant l'agrégation des réseaux. Un exemple sera présenté au chapitre 9.2.

3.2.4 Les adresses multicast

Les adresses multicast servent à atteindre plusieurs hôtes précis. Un paquet, envoyé depuis le serveur multicast, sera cloné par les routeurs uniquement lorsqu'ils devront l'envoyer par deux interfaces différentes. Ceci économisera beaucoup de bande passante. En IPv4, si deux utilisateurs du même sous-réseau regardent le même live-streaming sur leur ordinateur respectif depuis internet, il y aura deux flux de données partant du serveur. Ces deux flux effectueront exactement le même trajet en parallèle ce qui est totalement inutile. Un des nombreux buts d'IPv6 est d'améliorer cela et de ce fait, le multicast devient obligatoire.

Les adresses multicast ont toutes un préfix commun qui est « ff00::/8 ». Les bits de 9 à 12 sont appelés drapeaux et servent, entre autre, à dire si l'adresse est permanente ou pas. Les bits de 13 à 16 servent eux à définir la portée de l'adresse (locale, de site, d'organisation ou globale). Sur la Figure 1, ci-dessus, se trouvent deux de ces adresses. Comme on le verra plus loin, l'adresse « ff02::2 » sert à joindre tous les routeurs du lien (du sous-réseau). Les adresses avec le préfix « ff02::1:ff00:0000/104 » atteignent tous les hôtes qui ont une adresse se terminant par les mêmes 24 bits.

3.3 L'adressage

Le plan d'adressage en IPv6 ne change pas beaucoup de celui d'IPv4 dans la conception. Les sous-réseaux sont toujours construits avec la même logique tout en essayant de garder une hiérarchie maximale mais vu que l'on a beaucoup plus de bits à disposition, il faut les utiliser. Comme disait Mme Hagen : « Si une personne qui s'est occupée de l'adressage en IPv4 ne sent pas ses poils s'hérisser lorsqu'elle fait un plan d'adressage IPv6, c'est qu'elle ne donne pas assez de sous-réseaux »[1] Par contre, étant donné que les masques sont tous les mêmes (/64), il ne faut plus se préoccuper de la taille de notre sous-réseau. Les liaisons point-à-point et les adresses de loopback sont les seules exceptions sur lesquels on va mettre des préfixes plus grands : /126, respectivement /128.

Comme vu au point 3.2, il existe différents types d'adresses. Le plan d'adressage se fait uniquement avec des adresses de site ou globales. Si on devait les mettre les deux, ils seraient construits pareillement. On peut, par contre, avec IPv6 se permettre de ne mettre les adresses de liaisons point-à-point que dans un réseau (celui de management) lorsque l'on en a plusieurs dans le cas d'utilisation de VRF (cf. chapitre 8) ou si l'on utilise des adresses de site et des adresses globales simultanément. Ces adresses de liaisons point-à-point servent uniquement à savoir si les interfaces fonctionnent correctement mais les routeurs utilisent, entre eux, les adresses de liaisons-locales.

3.4 IPv6, comment transiter ?

Le passage à IPv6 va forcément devoir se faire en douceur en commençant, par créer un réseau de test avec les mêmes caractéristiques que celui que l'on veut transformer, donc un réseau IPv4. Sur celui-ci, on va intégrer IPv6 comme on désire le faire ensuite dans le réseau de production. Pour se faire, il existe plusieurs méthodes de cohabitation qui sont expliquées ci-dessous.

3.4.1 Dual-stack

Le dual-stack est la technique la plus simple. Elle consiste à rajouter la pile IPv6 à côté de celle d'IPv4 ainsi, un hôte peut utiliser soit l'un soit l'autre lors de l'envoi de paquets. La seule restriction à cette

méthode est que le protocole choisit pour l'envoi d'un paquet doit être actif jusqu'à la destination car elle ne permet pas le changement de protocole. Le but est que tous les hôtes, qui peuvent l'implémenter, utilisent IPv6 entre eux et qu'ils se basent encore sur IPv4 pour les machines qui n'ont pas la fonction dual-stack (on pense particulièrement à des équipements comme les imprimantes réseaux qui sont souvent vieux mais qui fonctionnent toujours). Avec cette méthode, on fera donc cohabiter les deux protocoles tant que l'on utilise IPv4 et, seulement une fois que l'on est certain de pouvoir s'en passer, on enlèvera la pile IPv4. Il est important de remarquer que cela demandera une charge bien plus grande aux équipements réseaux car ils devront être capables de tout faire à double et d'avoir entre-autre deux tables de routage distinctes. Le travail de M. Lienhard explique comment un hôte choisit quel protocole utiliser. Les équipements réseaux, tant qu'ils sont configurés pour le faire, s'adaptent au choix de l'hôte.

Dans notre cas, au SIEN, il a été décidé d'opter pour le dual-stack. Nous allons donc, sur nos prototypes, implémenter IPv4 puis, comme en cohabitation, installer IPv6 à côté. Cette manière de faire ne doit, dans la théorie, pas avoir une influence sur le réseau IPv4 déjà présent et on peut sélectivement choisir quel protocole on veut utiliser. De plus, il sera facile d'observer le comportement du protocole et de ses applications directes comme le routage.

Comme l'a expliqué Mme Hagen lors de sa présentation, il existe deux grandes approches pour la transition dual-stack : Soit « core-to-edge » soit « edge-to-core ». La première, qui propose de commencer par le centre du réseau avant de se diriger vers les extrémités est conseillée lorsque le temps le permet et est moins couteuse. La seconde approche, qui est l'opposée de la première, s'utilise dans les cas où IPv6 doit être implémenté rapidement. Cette solution est certainement plus coûteuse et complexe[5]. De plus, tant que le cœur du réseau ne permet pas de faire transiter IPv6, il va être difficile d'utiliser une application qui doit se trouver à plusieurs endroits physiques.

3.4.2 Tunnel IPv6-sur-IPv4

Le dual-stack n'est fait que pour transiter de bout-en-bout sur un seul protocole. Cela fonctionne très bien si un des protocoles est implémenté de bout-en-bout mais peut poser des problèmes au moment où l'on devra passer « à travers » IPv4 pour atteindre notre destination IPv6. Imaginons une entreprise se situant sur deux sites. Pour faire joindre les deux sites, il faudrait que IPv6 soit routé entre les deux ce qui n'est pas forcément le cas. Dans cette situation il est donc possible de mettre en place un tunnel entre le routeur du premier site et celui du second comme le montre la Figure 2. Une trame IPv6, devant transiter entre les deux, sera encapsulée dans un paquet IPv4 par le routeur avec comme destination l'adresse IPv4 de l'autre site. Une fois arrivée, le routeur enlève l'encapsulation et envoie la trame sur son réseau interne. Un des protocoles utilisé pour implémenter cette méthode s'appelle « 6in4 ».

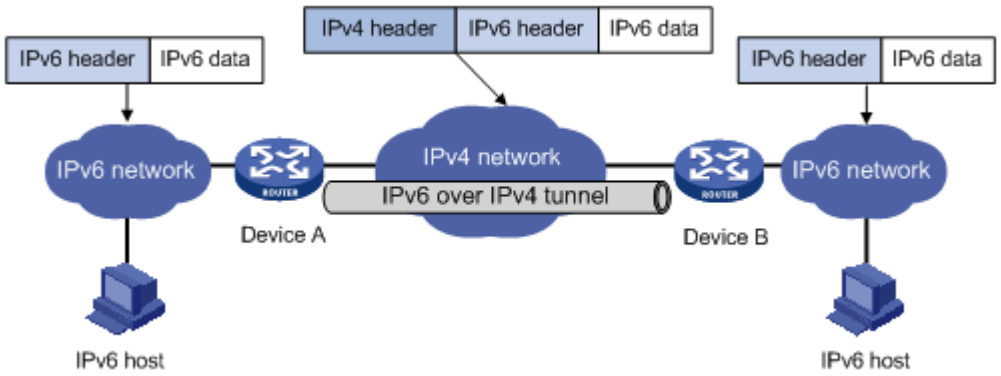

Figure 2 : Tunnel IPv6-sur-IPv4 (source : h3c.com)

Une autre version du tunnel IPv6-sur-IPv4 utilise des adresses IPv6 spéciales qui contiennent l'adresse IPv4 de destination. Le routeur à la sortie du réseau IPv6 peut donc encapsuler la trame IPv6 dans une trame IPv4 avec directement la bonne adresse de destination. La machine de l'autre côté, qui implémente le dual-stack, va donc désencapsuler le paquet qu'elle a reçu et le traiter ensuite comme s'il avait transité tout le long en IPv6 comme le montre la Figure 3. Cette méthode est appelée officiellement « Teredo » ou encore « tunnel automatique », car il ne faut rien configurer de statique alors que pour la première, il faut assigner, en manuel et sur chaque routeur, une adresse IPv4 pour tous les réseaux IPv6 que l'on veut joindre ce qui demande beaucoup de travail à chaque modification.

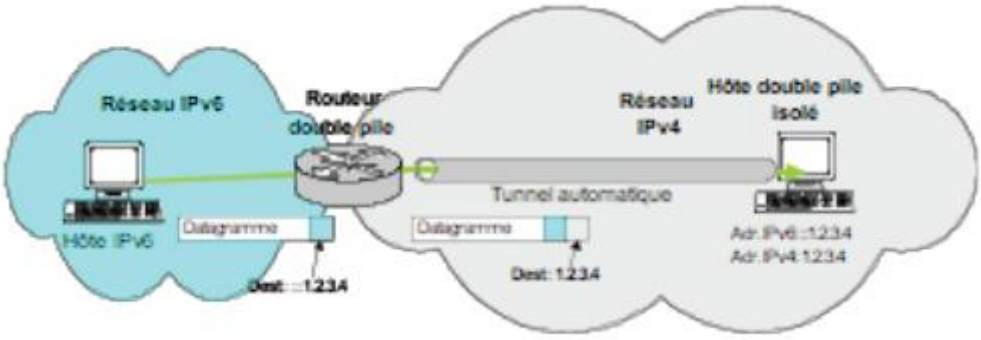

Figure 3 : Tunnel automatique IPv6-sur-IPv4 (source : cours PDR 2010)

3.4.3 Tunnel IPv4-sur-IPv6

Le tunnel IPv4-sur-IPv6 fonctionne exactement à l'inverse de celui d'IPv6-sur-IPv4. C'est-à-dire qu'il servira à faire transiter des réseaux IPv4 qui n'ont pas encore migrés à travers le réseau IPv6. Ce sont aussi les routeurs qui s'occuperont de faire l'encapsulation (IPv4 dans IPv6). Ces tunnels seront utilisés à la fin d'IPv4 ce qui veut dire dans très longtemps car IPv4 n'est pas près de disparaître comme on le verra ci-dessous.

3.5 IPv6, quand ?

Autant être clair tout de suite, l'utilisation d'IPv6 à la place d'IPv4 n'est pas prévue pour tout de suite pour un grand nombre de raisons. Premièrement IPv4 fonctionne encore très bien et les entreprises sont, pour l'instant, réticentes à investir dans les recherches d'autant plus que très peu d'ingénieurs réseaux sont au point dans ce domaine. Dans les entreprises toujours, tout matériel présent ne

supporte pas encore IPv6 de manière aussi sûre et rapide que c'est le cas avec IPv4. Deuxièmement, la transition en elle-même va durer un certain laps de temps : Une fois que l'on a pu analyser le fonctionnement du protocole et qu'il fonctionne sur le réseau de test, il va être possible de l'implémenter très gentiment sur de petites parties du réseau de production pour commencer et, quand on voit que cela ne pose pas de problème non-plus, continuer sur le reste, toujours par petites étapes. Après cela et pendant plusieurs années, les deux protocoles fonctionneront en parallèle jusqu'à ce que l'on remarque qu'IPv4 n'est plus utile et à ce moment, à nouveau par petites étapes, on pourra l'enlever. Mais cela est encore une musique d'avenir bien lointaine.

Ce qui pourrait faire accélérer la transition sont les applications qui demandent IPv6 car « Les clients ne demandent pas IPv6, mais des services qui demandent IPv6 »[2]. Cela implique que des personnes ne connaissant pas les réseaux voudront, un jour, pouvoir utiliser IPv6 très rapidement et à ce moment, les ingénieurs réseau devront trouver une solution rapide et donc intégrer IPv6.

4 Réalisation

4.1 Les prototypes

Comme la période de travail est divisée en deux parties, une première à raison d'un jour par semaine pendant 16 semaines et une deuxième à 100% pendant 6 semaines, il a été décidé de diviser le travail lui-même en deux étapes. Une première principalement pour découvrir les bases du protocole et une deuxième pour approfondir le sujet et se ramener aussi près que possible du réseau qui existe déjà. Ainsi nous allons travailler sur deux prototypes différents :

4.1.1 Prototype 1

Sur le prototype 1, nous avons fait un réseau simple qui va nous aider à comprendre les bases de l'IPv6. Ce réseau utilisera OSPF, avec une seule area, comme protocole de routage. Il comprendra trois zones d'hôtes et utilisera des switchs L2 et L2-L3, ainsi que différents routeurs et une connexion série. C'est un des montages les plus basiques qu'il est possible de faire tout en touchant à tout ce que demande un réseau (switching, routage, différents médias de transport). Ce premier prototype pourrait ensuite être intégré au deuxième comme une petite partie de celui-ci. Il est donc important qu'il fonctionne correctement avant de passer à la suite.

4.1.2 Prototype 2

Le prototype 2 a donc commencé avec la deuxième partie du travail. Il va, une fois que l'on a bien analysé les bases, nous aider à nous rapprocher le plus possible des spécifications du réseau du SIEN. Il sera donc bien plus complexe que le premier car le réseau sera plus grand et nous aurons aussi plus de matériel à disposition. Ce prototype nous indiquera si le matériel et la technologie sont prêts pour que le SIEN puisse commencer une transition sur IPv6 et dans le cas contraire, nous montrera ce qui n'est pas encore fonctionnel.

Contrairement à ce qui a été pensé initialement, le prototype 1 n'a pas été repris tel quel dans une partie de ce second prototype. Cela aurait été possible, mais au vu du nombre d'équipements déjà utilisés, il a été plus logique de séparer notre premier prototype pour le remettre, parties par parties, à plusieurs endroits.

Etant donné que tout n'a pas fonctionné comme espéré sur ce prototype, il a été simplifié afin de faire un montage fonctionnel et c'est pourquoi on trouvera un second plan d'adressage n'utilisant que le matériel compatible avec les options compatibles. Celui-ci sert uniquement à montrer le réseau que l'on a réussi à monter en IPv6.

4.2 Les plans d'adressage

4.2.1 Prototype 1

Etant donné que le prototype comporte trois zones d'hôtes, il y aura trois sous-réseaux de base en plus des liaisons point-à-point et des adresses de loopback. Comme pour IPv4, un bloc a été pris pour toutes les adresses de loopback, un autre pour toutes les liaisons point-à-point. La partie serveurs a aussi été séparée de la partie clients. Par contre, les blocs pris pour ces parties sont beaucoup plus grands que ce que l'on a en IPv4 ; à titre de comparaison, on donne des /48 à chaque partie ce qui laisse encore 16 bits pour diviser en sous-réseaux. 16 bits est ce que l'on a au total pour tout notre réseau IPv4, hôtes compris. Comme expliqué précédemment, on ne doit donc pas se soucier du futur

en réfléchissant à devoir laisser un grand nombre d'adresses libres ou non. Le plan d'adressage se trouve directement sur le schéma du réseau, Figure 4. Il comporte aussi le nombre de sous-réseaux restants pour se faire une idée plus claire de la grandeur des plages d'adresses.

4.2.2 Prototype 2

Ce deuxième plan d'adressage a été complétement révisé, on ne s'appuie donc pas sur le premier. Il a été élaboré pour un environnement beaucoup plus grand que celui du prototype et devrait pouvoir servir de base pour le SIEN.

Etant donné que nous avons plusieurs réseaux totalement différents et séparés (police, hôpitaux, administration), il faudra premièrement diviser le préfixe reçu afin que ces réseaux soient aussi clairement séparés dans l'adressage. Il semble logique de donner à chacun un préfix de 48 bits, taille d'un réseau reçu par une entreprise. Avec cette taille de préfix on sait que l'on peut créer 65'536 sous-réseaux ce qui sera largement suffisant dans notre cas. Dans chaque réseau, afin d'avoir une certaine hiérarchisation qui permettra de simplifier les tables de routage, les deux premiers nombres hexadécimaux sont utilisés pour le séparer en plusieurs zones et ceci aidera à mieux reconnaître les adresses lors de leur lecture. Cela laisse encore huit bits variables qui peuvent être utilisés pour séparer en catégories (Client, Serveur, Téléphone), puis pour faire des sous-réseaux dans celles-ci. Pour les grandes zones, on peut s'imaginer donner deux préfixes successifs vu le nombre que l'on peut en faire.

Pour résumé, l'adresse « 2001:4DA0:C01:34F8::2 » pourra être identifiée comme suit : « 2001 :4DA0 :C** » est le /40 reçu par le SIEN. Le « 01 » nous indique de quel réseau il s'agit (par exemple : administration). Le « 34 » indique la zone ou la région où se trouve l'hôte possédant cette adresse. Le « F » indique c'est un serveur et finalement le « 8 » sert à savoir dans quel sous-réseau est ce serveur. Les 64 derniers bits, donc le « 2 » final dans ce cas, définissent l'hôte en lui-même.

La Figure 5 nous montre le schéma et le plan d'adressage de ce deuxième prototype. On peut remarquer qu'il reste, même avec cet adressage pensé largement, encore un grand nombre grand de réseaux libres étant donné qu'il y a 256 « /48 » possibles. Donc là aussi, il est possible d'en donner deux (ou plus) consécutifs aux plus grands réseaux. Comme on l'a déjà vu, il ne faut vraiment pas hésiter à attribuer trop de place pour les différents réseaux.

La Figure 6 est le schéma et plan d'adressage du réseau simplifié qui a été utilisé pour faire fonctionner le backbone en multi-VRF (cf. chapitre 9.2).

4.2.3 Remarques

Comme on l'a vu, les adresses IPv6 sont bien plus longues et compliquées à retenir et même à écrire. De plus, l'idée d'IPv6 est de faire abstraction des adresses et de ne plus devoir travailler directement avec elles. L'utilisation du DNS (Domain Name System) est donc très importante. Si les hôtes peuvent s'y inscrire de manière assez simple (voir le travail de M. Lienhard), c'est un grand problème pour tous les équipements réseau. La technique pour pouvoir les atteindre rapidement et simplement est donc de les inscrire manuellement et statiquement dans le serveur DNS. Ainsi il suffit d'attaquer un équipement par son nom et l'on ne doit pas se soucier de son adresse. Cela n'enlève pas l'utilité de toujours devoir posséder une version papier des adresses de ces équipements en cas de panne du serveur DNS.

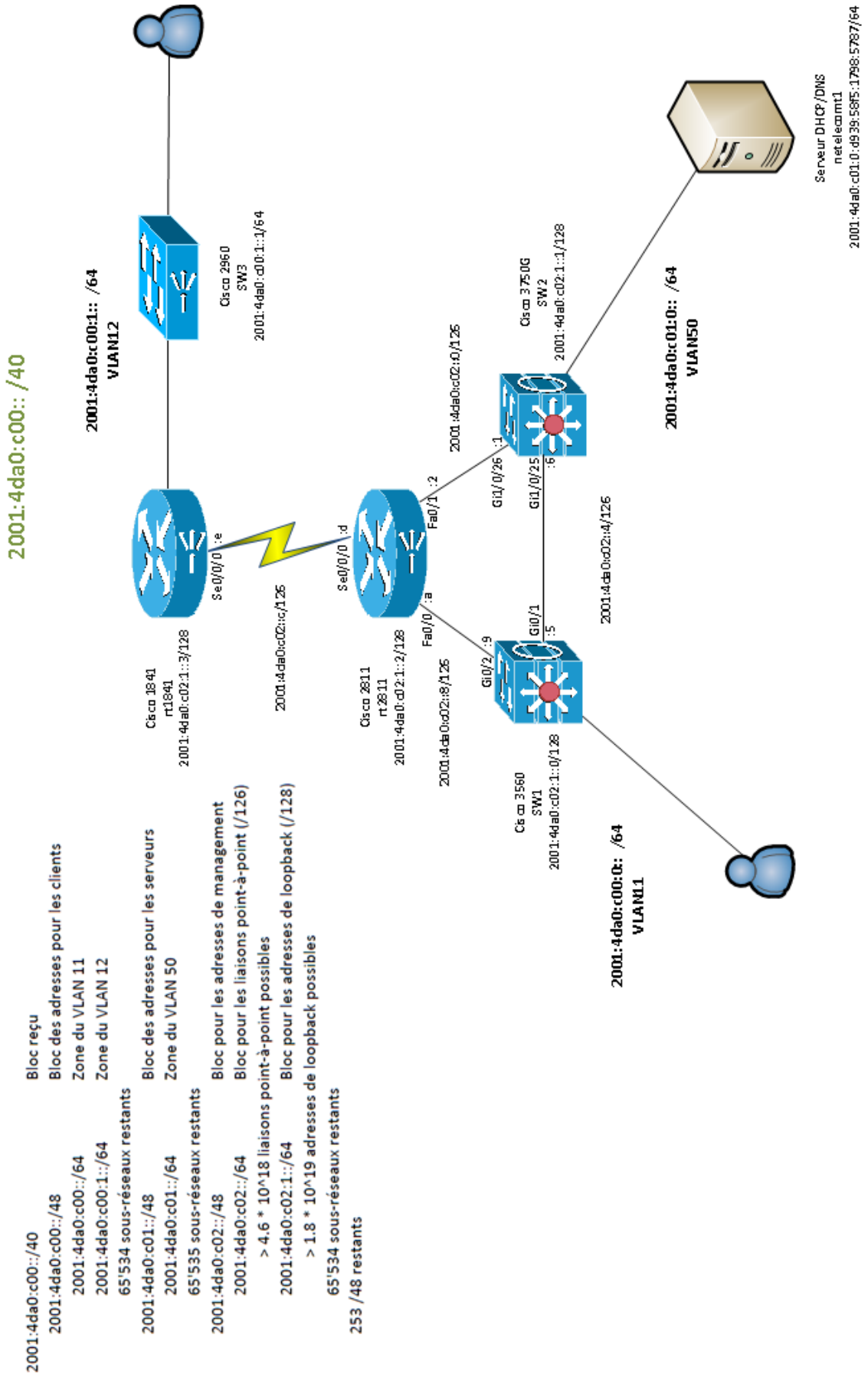

Figure 4 : Schéma et plan d'adressage du prototype 1

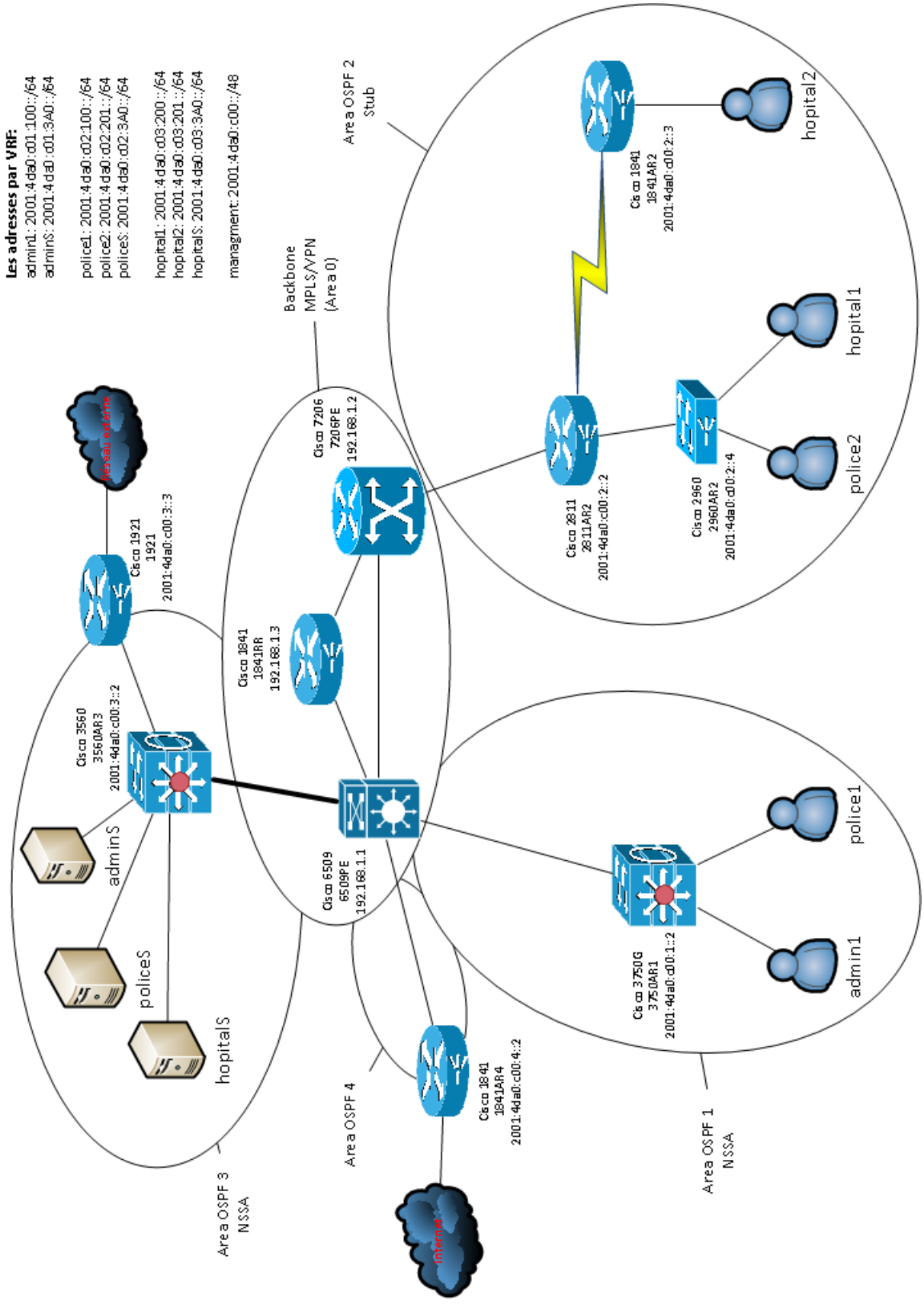

Figure 5 : Schéma et plan d'adressage du prototype 2 entier

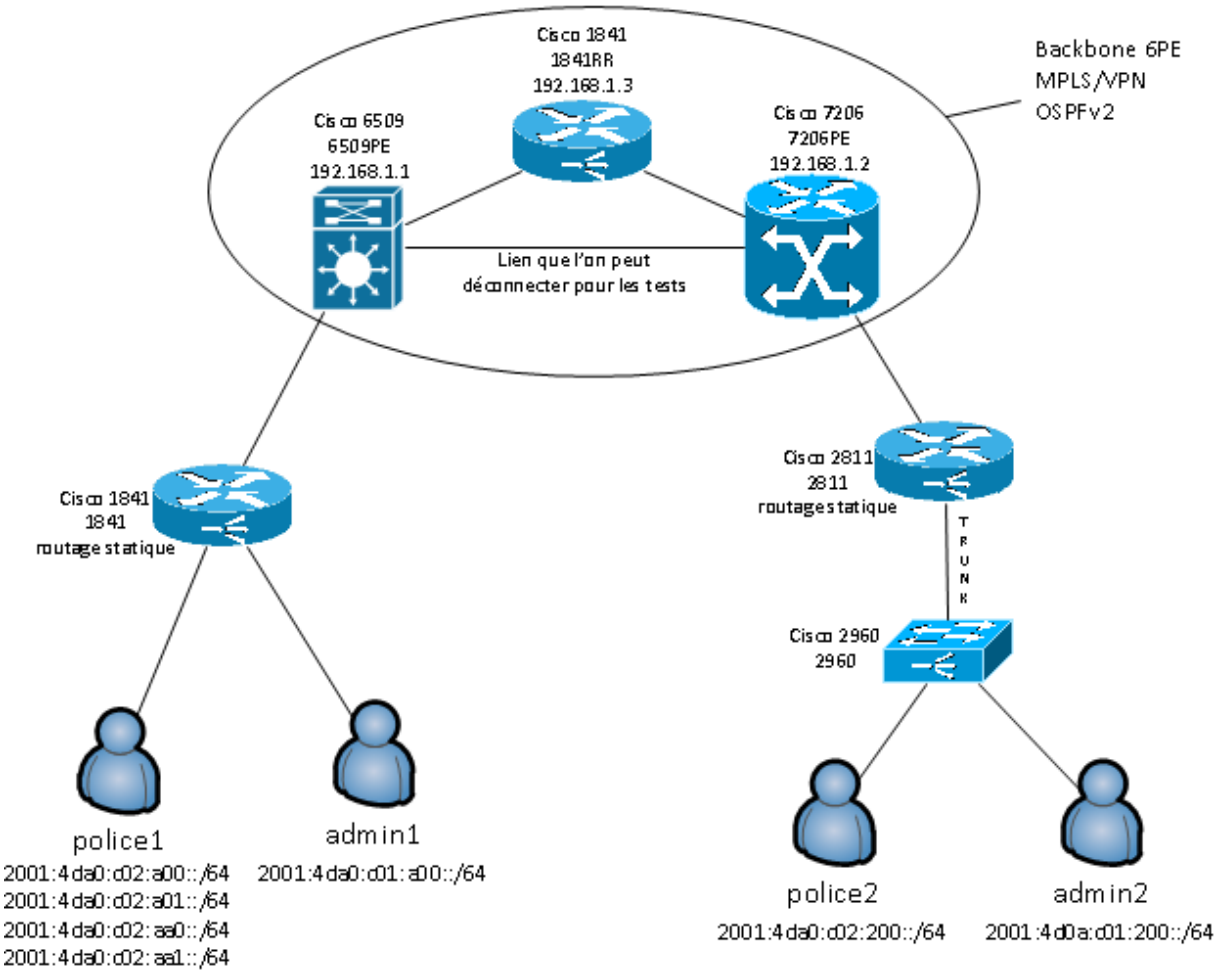

Figure 6 : Schéma et plan d'adressage du prototype 2 simplifié

4.3 Matériel utilisé

Tout le matériel utilisé pour ce travail est de marque Cisco. Pour le prototype 1, les deux switchs L2-L3 sont des WS-C3560-48PS-S et WS-C3750G-24TS-E1U, les deux routeurs sont des 2811 et 1841 et le switch L2 est un WS-C2960-8TC-L. C'est le matériel qui est le plus utilisé au SIEN. Le petit switch est souvent mis derrière un routeur 1841 qui lui est relié au centre du réseau par un modem. Les switchs L2-L3 sont eux utilisés pour éviter de devoir avoir deux appareils (un routeur et un switch L2) à certains endroits dans le réseau où un grand nombre de ports est nécessaires. On trouve une petite photo du montage dans les annexes (Images).

Pour le prototype 2, deux équipements principalement utilisés pour le backbone BGP/MPLS (cf. chapitre 10) ont été rajoutés. On aura donc en plus un nouveau switch L2-L3 (WS-C6509-E) et un routeur (7206VXR). D'autres équipements déjà utilisés lors du premier prototype ont été doublés ou triplés et un petit routeur 1921 a encore fait son apparition. Une image se trouve là-aussi dans les annexes (Images)

4.3.1 IOS

Il a été décidé de prendre les IOS (Internetwork Operating System) les plus avancés pour faire nos tests, c'est pourquoi sur les switchs nous avons pris les versions « ipservices » et sur les routeurs

« advanced ipservices ». Ces versions offrent le plus d'options possibles et permettent ainsi de voir toutes les possibilités existantes.

Pour le prototype 1, au début du travail, les IOS les plus récents ont été installés et gardés jusqu'à la fin de ce prototype. Pour le second, étant donné que certaines fonctions utiles n'étaient pas encore implémentées, un contrôle a été effectué régulièrement afin de pouvoir utiliser les toutes dernières fonctionnalités en permanence tout en espérant que celles que l'on désire sortent. Malheureusement il n'y a pas eu de mise-à-jour utile pendant la période du travail. Le Tableau 1 montre quels IOS ont été utilisés à quel moment.

	prototype	
équipement	1	2
switch 2960	12.2.55-SE1 lanbase	12.2.58-SE1 lanbase
switch L2-L3 3560	12.2.55-SE1 ipservices	12.2.55-SE3 ipservices
switch L2-L3 3750G	12.2.55-SE1 ipservices	12.2.58-SE1 ipservices
switch L2-L3 6509	-	12.2.33-SXJ advipservices
routeur 1841	15.1.3T advipservices	15.1.4M advipservices
routeur 1921	-	15.1.4M1 universal
routeur 2811	15.1.3T advipservices	15.1.4M advipservices
routeur 7206	-	15.1.4M advipservices

Tableau 1 : Version des IOS utilisées par équipement pour chaque prototype

Les versions terminées par un « T » sont déconseillées pour l'utilisation en production car elles ne sont pas encore déclarées stables. Par contre, elles comprennent un grand nombre d'options supplémentaires (surtout pour IPv6) mises-à-jour régulièrement et que celles de production comprendront aussi dans le futur. Vu que l'intégration d'IPv6 au réseau de production ne se fera pas immédiatement, il n'y a aucun problème à travailler sur ces versions. De plus, les versions « T » utilisées pour le premier prototype sont passées en « M » avant le début du deuxième. Les versions « M » doivent être plus stables et leurs mises-à-jour ne contiennent pas de nouvelles fonctionnalités mais uniquement des corrections de beugs.

4.4 Outils

4.4.1 Wireshark

L'outil le plus utilisé pour capturer et analyser les paquets transitant sur un réseau est « Wireshark » et c'est avec lui que toutes les captures de paquets présentées dans ce travail sont effectuées. De plus, les routeurs ont une fonction pour capturer les paquets entrant ou sortant d'une interface et de les enregistrer au format « pcap » qui est celui lu par « Wireshark ». La version du programme utilisée est la 1.4.6.

4.4.2 Putty

Pour se connecter aux équipements réseaux, c'est le petit programme « Putty » qui a été utilisé. Il permet autant de se connecter en telnet que par le câble console. De plus, on peut aussi facilement enregistrer, grâce à lui, tout ce que l'on envoie et reçoit dans un fichier texte. La version utilisée est la 0.60.

4.4.3 PRTG Network Monitor

Pour tester les performances des différents équipements, le programme PRTG (Paessler Router Traffic Grapher) a été utilisé. Il permet d'aller rechercher différentes informations grâce au protocole SNMP. Le programme, installé sur le serveur, va faire des requêtes régulières (toutes les 30 secondes) et proposer des graphiques sur différentes périodes de temps. Il pourrait aussi lancer des alarmes par e-mail dans certains cas critiques si on le désire. La version utilisée est la 8.4.2.2356.

Comme on le verra, nous avons capturé les informations des CPU, des interfaces et de la mémoire RAM. Pour cette dernière uniquement, les captures ont été effectuées toutes les 5 minutes.

5 Activation du dual-stack

Les routeurs ainsi que les switchs L2-L3 avec les dernières versions d'IOS sont prêts à implémenter la double pile IPv4-IPv6. Par contre, le processus pour activer cette fonction est plus compliqué sur les switchs. On va donc étudier la partie spéciale de ces derniers en premier.

5.1 Switchs L2-L3

Les switchs L2-L3 possèdent plusieurs templates qui décident où mettre les priorités entre le switching et le routage. Ces templates sont appelés SDM (Switch Database Management). En IPv4, le choix est entre les modèles « switching uniquement », « switching et routage avec priorité sur le switching » (mode par défaut) et « switching et routage avec priorité au routage ». La différence entre ces modèles est l'espace mémoire attribué au stockage des adresses MAC ou des réseaux IP pour le routage. Pour qu'un switch L2-L3 fonctionne avec la double-pile, il faut utiliser un SDM « dual-ipv4-and-ipv6 » et ensuite choisir, comme pour IPv4, un mode de priorité. Cela se fait avec les commandes « sdm prefer dual-ipv4-and-ipv6 <mode> ». Les modes sont les mêmes qu'en IPv4. Ensuite, la partie L3 du switch s'utilise comme un routeur (cf. 5.2).

Il est important de noter que pour chaque changement de SDM, il faut redémarrer le switch. Sur ceux de production, il faut donc faire attention à quel moment intervenir. Comme on le verra ci-dessous, il faut aussi faire très attention à la place mémoire nécessaire pour la table de routage car la place utilisable va être plus petite avec la double-pile étant donné qu'elle doit comprendre deux tables de routage.

5.2 Routeur

Pour qu'une interface monte un stack IPv6, il faut lui donner une adresse IPv6. Cela peut se faire de deux manières : Avec la commande « ipv6 enable », elle va uniquement se construire une adresse de liaison-locale. Avec la commande « ipv6 address <adresse/masque> » on donne une adresse IPv6 à l'interface. Comme on l'a vu en théorie, dès que le stack est monté l'interface se construit une adresse de liaison locale, la première commande n'est donc pas obligatoire mais fortement conseillée. Si elle n'est pas dans la configuration et que l'on désire changer l'adresse IP insérée avec la commande « ipv6 address <…>» en commençant par « no » suivit de l'adresse à supprimer, on perd toutes les configurations IPv6 sur l'interface car on supprime le stack. Avec la commande « ipv6 enable » toujours présente, on ne prend pas ce risque.

Une fois que le stack est monté, l'interface agit comme l'interface d'un hôte et pas d'un routeur, on peut donc la pinguer ou se connecter dessus en telnet mais elle n'envoie pas de « Router Advertisement» (cf.6.4). Pour qu'elle le fasse, il faut encore entrer la commande suivante en mode de configuration globale : « ipv6 unicast-routing ». Une fois que ces deux étapes ont été effectuées, le routeur fonctionne comme on l'attend et gère autant les paquets IPv4 que les paquets IPv6.

5.3 Gestion mémoire du dual-stack

5.3.1 Sur les switchs

Le routage sur les switchs est géré par du hardware et on l'appelle la plateforme tcam. Les différents templates changent la répartition de la mémoire entre les différentes fonctions du switch. La Figure 7 nous montre comment la mémoire est répartie en mode par défaut. On voit que l'on peut avoir

environ 2'000 adresses MAC ainsi que 1'000 routes IPv4 et 1'000 IPv6. La Figure 8 nous montre, elle, le mode routing. On voit ici qu'il ne reste plus que 1'500 entrées pour les adresses MAC mais par contre 1'250 pour les routes IPv4 comme IPv6. D'autres fonctionnalités de switching sont aussi plus limitées avec ce mode.

```
sw1#sh sdm prefer dual-ipv4-and-ipv6 default
 "desktop IPv4 and IPv6 default" template:
 The selected template optimizes the resources in
 the switch to support this level of features for
 8 routed interfaces and 1024 VLANs.

  number of unicast mac addresses:                  2K
  number of IPv4 IGMP groups + multicast routes:    1K
  number of IPv4 unicast routes:                    3K
    number of directly-connected IPv4 hosts:        2K
    number of indirect IPv4 routes:                 1K
  number of IPv6 multicast groups:                  1.125k
  number of directly-connected IPv6 addresses:      2K
  number of indirect IPv6 unicast routes:           1K
  number of IPv4 policy based routing aces:         0
  number of IPv4/MAC qos aces:                      0.5K
  number of IPv4/MAC security aces:                 1K
  number of IPv6 policy based routing aces:         0
  number of IPv6 qos aces:                          0.625k
  number of IPv6 security aces:                     0.5K
```

Figure 7 : Template dual-stack par défaut

```
sw2#show sdm prefer dual-ipv4-and-ipv6 routing
 "desktop IPv4 and IPv6 routing" template:
 The selected template optimizes the resources in
 the switch to support this level of features for
 8 routed interfaces and 1024 VLANs.

  number of unicast mac addresses:                  1.5K
  number of IPv4 IGMP groups + multicast routes:    1K
  number of IPv4 unicast routes:                    2.75K
    number of directly-connected IPv4 hosts:        1.5K
    number of indirect IPv4 routes:                 1.25K
  number of IPv6 multicast groups:                  1.125k
  number of directly-connected IPv6 addresses:      1.5K
  number of indirect IPv6 unicast routes:           1.25K
  number of IPv4 policy based routing aces:         0.25K
  number of IPv4/MAC qos aces:                      0.5K
  number of IPv4/MAC security aces:                 0.5K
  number of IPv6 policy based routing aces:         0.25K
  number of IPv6 qos aces:                          0.625k
  number of IPv6 security aces:                     0.5K
```

Figure 8 : Template dual-stack orienté routing

Les indications vues ci-dessus étant uniquement théoriques, il existe une autre commande pour voir l'état présent avec la configuration actuelle. Cette commande va interroger la plateforme tcam pour nous donner des informations sur l'espace mémoire dédié : « show platform tcam utilization ». La Figure 9 nous montre le retour de la commande lorsque l'on utilise le template dual-stack par défaut.

On voit de grandes différences avec les nombres donnés sur la Figure 7 car, comme expliqué, la taille réelle de la mémoire disponible pour les adresses dépend de la configuration. Dans notre cas, les switchs n'ont qu'un VLAN et très peu d'options activées ce qui augmente l'espace mémoire pour le reste. La Figure 10 nous montre les mêmes informations mais avec le template routing activé. On voit que l'on a, comme nous l'indiquait le template, plus de place pour les routes et moins pour les adresses MAC. Finalement, point très intéressant, on peut comparer ces chiffres avec ceux d'un switch en place dans le réseau de production comme nous le montre la Figure 11. On voit que le nombre de routes IPv4 dépasse les 2'000 ce qui ne serait possible avec aucun template dual-stack. Donc si l'on décidait de changer le SDM sur un de ces switch, on risque d'avoir un gros problème et de faire tomber une partie du réseau. Cela implique que le passage au dual-stack ne pourra pas se faire avant d'avoir des switchs avec une capacité plus grande.

```
CAM Utilization for ASIC# 0                  Max              Used
                                             Masks/Values     Masks/values

 Unicast mac addresses:                       672/5376         21/65
 IPv4 IGMP groups:                            152/1216          6/26
 IPv4 unicast directly-connected routes:      672/5376         21/65
 IPv4 unicast indirectly-connected routes:    144/1152         15/65
 IPv6 Multicast groups:                       672/5376         21/65
 IPv6 unicast directly-connected routes:      672/5376         21/65
 IPv6 unicast indirectly-connected routes:    128/1024         12/44
 IPv4 policy based routing aces:                0/0             0/0
 IPv4 qos aces:                               768/768         260/260
 IPv4 security aces:                         1024/1024         39/39
 IPv6 policy based routing aces:                0/0             0/0
 IPv6 qos aces:                                 0/0             0/0
 IPv6 security aces:                          204/510           5/5

Note: Allocation of TCAM entries per feature uses
a complex algorithm. The above information is meant
to provide an abstract view of the current TCAM utilization
```

Figure 9 : Utilisation tcam avec le mode par défaut

```
CAM Utilization for ASIC# 0                  Max              Used
                                             Masks/Values     Masks/values

 Unicast mac addresses:                       544/4352         21/65
 IPv4 IGMP groups + multicast routes:         152/1216          6/26
 IPv4 unicast directly-connected routes:      544/4352         21/65
 IPv4 unicast indirectly-connected routes:    176/1408         15/65
 IPv6 Multicast groups:                       544/4352         21/65
 IPv6 unicast directly-connected routes:      544/4352         21/65
 IPv6 unicast indirectly-connected routes:    262/2096         12/44
 IPv4 policy based routing aces:              256/256           2/2
 IPv4 qos aces:                               768/768         324/324
 IPv4 security aces:                          512/512          39/39
 IPv6 policy based routing aces:                0/0             0/0
 IPv6 qos aces:                                 0/0             0/0
 IPv6 security aces:                          204/510           5/5

Note: Allocation of TCAM entries per feature uses
a complex algorithm. The above information is meant
to provide an abstract view of the current TCAM utilization
```

Figure 10 : Utilisation tcam avec le mode routing

```
CAM Utilization for ASIC# 0                    Max                  Used
                                               Masks/Values         Masks/values

 Unicast mac addresses:                         400/3200             133/974
 IPv4 IGMP groups + multicast routes:           152/1216             10/45
 IPv4 unicast directly-connected routes:        400/3200             133/974
 IPv4 unicast indirectly-connected routes:      1040/8320            278/2099
 IPv4 policy based routing aces:                384/512              1/2
 IPv4 qos aces:                                 768/768              388/388
 IPv4 security aces:                            1024/1024            33/33

Note: Allocation of TCAM entries per feature uses
a complex algorithm. The above information is meant
to provide an abstract view of the current TCAM utilization
```

Figure 11 : Utilisation tcam sur un switch de production

5.3.2 Sur les routeurs

Les routeurs utilisent du software et directement leur mémoire RAM pour le routage. Ils n'ont donc pas de plateforme tcam. Ainsi la taille maximale de la table de routage dépend de la mémoire RAM. En analysant ce qui se passait en ajoutant des routes IPv4 et IPv6 on peut faire les constatations suivantes : En IPv4 il faut compter un ordre de grandeur de 215 octets par réseau puis encore 200 octets par sous-réseau alors qu'en IPv6 on ne compte seulement 124 octets par préfix. On voit donc que les tables de routage prennent moins de place en IPv6 qu'en IPv4 pour le même nombre de routes comme le montre la Figure 12 (IPv4) et la Figure 13 (IPv6). Sachant que l'on peut faire de plus grandes agrégations de réseaux en IPv6, ces tables prennent donc vraiment moins de place que celles d'IPv4.

```
rt1841#sh ip route summary
IP routing table name is default (0x0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Replicates  Overhead    Memory (bytes)
connected       0           5           0           300         860
static          0           5           0           300         860
ospf 1          0           8           0           480         1408
  Intra-area: 8 Inter-area: 0 External-1: 0 External-2: 0
  NSSA External-1: 0 NSSA External-2: 0
internal        1                                               1104
Total           1           18          0           1080        4232
```

Figure 12 : Taille de la table de routage IPv4

```
rt1841#sh ipv route summary
IPv6 routing table name is default(0) global scope - 19 entries
IPv6 routing table default maximum-paths is 16
Route Source    Networks    Overhead    Memory (bytes)
connected       2           176         248
local           4           352         496
ND              0           0           0
ospf 1          8           704         992
  Intra-area: 8  Inter-area: 0  External-1: 0  External-2: 0
  NSSA External 1: 0  NSSA External 2: 0
static          5           440         620
  Static: 5  Per-user static: 0
Total           19          1672        2356
```

Figure 13 : Taille de la table de routage IPv6

5.4 Commandes

Les commandes pour charger le bon template sont présentées au point 5.4.1 alors que celles utiles à l'activation du dual-stack et de la gestion des adresses sur les interfaces pour les routeurs et les switchs L2-L3 se trouvent au point 5.4.2. Ici, le 6509 se comporte comme un routeur.

5.4.1 Switch L3

conf t	
sdm prefer dual-ipv4-and-ipv6 <mode>	Les différents modes sont: -'**default**' pour routing et switching -'**routing**' avec plus de place mémoire pour le routing -'**vlan**' uniquement switching demande un redémarrage
end	
reload	Obligatoire lors d'un changement de sdm
show sdm prefer	Pour voir le mode actuel
show sdm prefer dual-ipv4-and-ipv6 <mode>	Pour voir les caractéristiques du mode : -'**default**' pour routing et switching -'**routing**' avec plus de place mémoire pour le routing -'**vlan**' uniquement switching
show platform tcam utilization	Pour voir l'état actuel de la mémoire tcam

5.4.2 Routeur et Switch L3

conf t	
ipv6 unicast-routing	Active la transmission des paquets en IPv6
interface <interface>	
ipv6 enable	Active IPv6 sur l'interface => elle va se créer une adresse de liaison locale
ipv6 address <prefix>::/<longeurPrefix> eui-64	Active IPv6 sur l'interface et indique un préfix avec lequel une adresse va se créer en fonction de l'adresse MAC
end	

show ipv6 interface brief	Permet de voir les interfaces qui sont activées en IPv6 et leurs adresses
show ipv6 route summary	Permet de voir la place en mémoire prise par la table de routage

6 ICMPv6

Comme expliqué précédemment, les différences entre IPv4 et IPv6 ne sont pas seulement l'extension de la plage d'adresse mais aussi l'amélioration et l'ajout de différentes fonctions. L'une des plus grandes est la mise à niveau du protocole ICMP (Internet Control Message Protocol) avec l'ajout de nouveaux types. Grace à ce nouveau protocole, ARP (Address Resolution Protocol) a été remplacé par le NDP (Neighbor Discovery Protocol) qui utilise cinq types d'ICMPv6 pour découvrir ses voisins. Ce chapitre analyse ces différents types. Il faut noter qu'ICMPv6 a gardé les autres fonctionnalités comme l'envoi de messages d'erreur mais que la gestion du MTU (Maximum Transmission Unit) a été modifiée et les routeurs ne vont plus fragmenter les paquets comme c'est le cas en IPv4. Cela sera aussi traité dans ce chapitre.

6.1 Neighbor Solicitation

Comme on l'a déjà vu au point 3.2.1, le « Neighbor solicitation » est le premier paquet envoyé par le hôte afin qu'il puisse confirmer que son adresse de liaison locale soit unique sur le lien. Mais, le but premier de ce type d'ICMPv6 est de connaître l'adresse de couche 2 correspondant à une IP(v6) comme le fait ARP pour IPv4. En ne recevant aucune réponse, l'hôte peut considérer que cette adresse est libre comme ce fut le cas sur la Figure 1. Ce type d'ICMP peut aussi être utilisé pour s'assurer qu'un hôte est toujours connecté. Dans ce cas il enverra le paquet à une adresse unicast alors que dans les autres cas ce sera une adresse multicast. La réponse aux « Neighbor solicitation » est un « Neighbor advertisement ».

Sur la Figure 14, on voit le champ « Type » qui indique à quel ICMPv6 on a à faire puis le champ « Target » qui donne l'adresse IPv6 de l'hôte dont on cherche les informations. On voit aussi que les adresses de destination de couches 2 et 3 sont des adresses multicast réservées pour IPv6.

```
⊞ Ethernet II, Src: Apple_a9:ba:04 (00:25:4b:a9:ba:04), Dst: IPv6mcast_ff:bc:ae:62 (33:33:ff:bc:ae:62)
⊞ Internet Protocol Version 6, Src: fe80::ac90:758c:83fc:4c9c (fe80::ac90:758c:83fc:4c9c), Dst: ff02::1:ffbc:ae62 (ff02::1:ffbc:ae62)
⊟ Internet Control Message Protocol v6
    Type: 135 (Neighbor solicitation)
    Code: 0
    Checksum: 0x8d31 [correct]
    Reserved: 0 (Should always be zero)
    Target: fe80::b6a4:e3ff:febc:ae62 (fe80::b6a4:e3ff:febc:ae62)
  ⊟ ICMPv6 Option (Source link-layer address)
      Type: Source link-layer address (1)
      Length: 8
      Link-layer address: 00:25:4b:a9:ba:04
```

Figure 14 : ICMPv6 Neighbor Solicitation

6.2 Neighbor Advertisement

Le « Neighbor advertisement » est la réponse à un « Neighbor solicitation ». Il sert à donner des informations comme le « rôle » joué (si c'est un routeur) et surtout l'adresse de couche 2. Les réponses aux « Neighbor solicitation » sont envoyées en unicast mais ce type peut aussi être envoyé en multicast afin d'annoncer un changement (de rôle ou d'adresse de couche 2).

Sur la Figure 15, on voit en plus du type qui a changé, les « Flags » qui permettent de donner quelques informations. Ici, le « Neighbor Advertisement » a été demandé (*Solicited*), il remplace les anciennes annonces (*Override*) et ce n'est pas un routeur qui l'envoie. Il faut noter que la « Target » est ici la source car c'est le sujet du paquet. Finalement, l'information désirée se trouve à la dernière ligne : C'est l'adresse de couche 2 de la « Target ».

```
⊞ Ethernet II, Src: Apple_a9:ba:04 (00:25:4b:a9:ba:04), Dst: Cisco_bc:ae:62 (b4:a4:e3:bc:ae:62)
⊞ Internet Protocol Version 6, Src: fe80::ac90:758c:83fc:4c9c (fe80::ac90:758c:83fc:4c9c), Dst: fe80::b6a4:e3ff:febc:ae62 (fe80::b6a4:e3ff:febc:ae62)
⊟ Internet Control Message Protocol v6
    Type: 136 (Neighbor advertisement)
    Code: 0
    Checksum: 0xe71d [correct]
  ⊟ Flags: 0x60000000
      0... .... .... .... .... .... .... .... = Not router
      .1.. .... .... .... .... .... .... .... = Solicited
      ..1. .... .... .... .... .... .... .... = Override
    Target: fe80::ac90:758c:83fc:4c9c (fe80::ac90:758c:83fc:4c9c)
  ⊟ ICMPv6 Option (Target link-layer address)
      Type: Target link-layer address (2)
      Length: 8
      Link-layer address: 00:25:4b:a9:ba:04
```

Figure 15 : ICMPv6 Neighbor Advertisement

6.3 Router Solicitation

Comme on l'a déjà vu au point 3.2.1, le « Router solicitation » est le deuxième paquet envoyé par un hôte IPv6 lorsqu'il se connecte. Le but de ce paquet est, comme son nom l'indique, de trouver les routeurs qui sont sur le même lien que lui afin d'y récupérer diverses informations comme le (ou les) préfix(es), la passerelle par défaut ou encore si on a besoin de joindre un serveur DHCP. Ces paquets sont toujours envoyés en multicast à l'adresse « ff02::2 » qui est celle pour joindre tous les routeurs. La réponse à un « Router solicitation » est un « Routeur advertisement » et bien que les routeurs envoient à intervalles réguliers ces derniers, un « Router solicitation » force à l'envoi immédiat. La Figure 16 nous montre comment est construit ce paquet. On remarque que, excepté le champ « Target » qui n'existe plus, il ressemble fortement au « Neighbor Solicitation ».

```
⊞ Ethernet II, Src: Apple_a9:ba:04 (00:25:4b:a9:ba:04), Dst: IPv6mcast_00:00:00:02 (33:33:00:00:00:02)
⊞ Internet Protocol Version 6, Src: fe80::ac90:758c:83fc:4c9c (fe80::ac90:758c:83fc:4c9c), Dst: ff02::2 (ff02::2)
⊟ Internet Control Message Protocol v6
    Type: 133 (Router solicitation)
    Code: 0
    Checksum: 0x83a5 [correct]
  ⊟ ICMPv6 Option (Source link-layer address)
      Type: Source link-layer address (1)
      Length: 8
      Link-layer address: 00:25:4b:a9:ba:04
```

Figure 16 : ICMPv6 Router Solicitation

6.4 Router Advertisement

Le « Router advertisement » est un paquet ICMPv6 envoyé régulièrement par les routeurs sur leurs liens. Ces paquets contiennent différentes informations comme le préfix de lien, le MTU maximum sur le lien, des routes spéciales, le mode DHCP (cf. chapitre 7) ou encore la durée pour laquelle une adresse est valable. La possibilité d'envoyer l'adresse du serveur DNS est une option en cours d'élaboration qui permettra de se passer encore plus des serveurs DHCP. Le routeur envoie ces paquets à l'adresse multicast « ff02::1 » qui atteint tous les hôtes du lien depuis son adresse de liaison-locale. Les « Routeur advertisement » doivent aussi être envoyés par les routeurs lorsqu'ils reçoivent un « Router solicitation ».

La Figure 17 montre un exemple typique de « Router advertisement ». Sous « Flags » sont les options d'utilisation du serveur DHCP comme on le verra au chapitre 7. Puis, en tant qu'options, se trouvent les différentes informations comme le préfix de lien, l'adresse de couche 2 du routeur, et la MTU du lien. On voit aussi l'adresse de destination du paquet qui est bien la multicast « ff02::1 » qui atteindra tous les hôtes du lien.

```
⊞ Ethernet II, Src: Cisco_bc:ae:62 (b4:a4:e3:bc:ae:62), Dst: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
⊞ Internet Protocol Version 6, Src: fe80::b6a4:e3ff:febc:ae62 (fe80::b6a4:e3ff:febc:ae62), Dst: ff02::1 (ff02::1)
⊟ Internet Control Message Protocol v6
    Type: 134 (Router advertisement)
    Code: 0
    Checksum: 0x153a [correct]
    Cur hop limit: 64
  ⊟ Flags: 0x40
      0... .... = Not managed
      .1.. .... = Other
      ..0. .... = Not Home Agent
      ...0 0... = Router preference: Medium
      .... .0.. = Not Proxied
    Router lifetime: 1800
    Reachable time: 0
    Retrans timer: 0
  ⊟ ICMPv6 Option (Source link-layer address)
      Type: Source link-layer address (1)
      Length: 8
      Link-layer address: b4:a4:e3:bc:ae:62
  ⊟ ICMPv6 Option (MTU)
      Type: MTU (5)
      Length: 8
      MTU: 1500
  ⊟ ICMPv6 Option (Prefix information)
      Type: Prefix information (3)
      Length: 32
      Prefix Length: 64
    ⊟ Flags: 0xc0
        1... .... = On-link flag(L): Set
        .1.. .... = Autonomous address-configuration flag(A): Set
        ..00 0000 = Reserved: 0
      Valid lifetime: 2592000
      Preferred lifetime: 604800
      Reserved
      Prefix: 2001:4da0:c00:1::
```

Figure 17 : ICMPv6 Router Advertisement

6.5 Redirect

Le « Redirect » est un type important pour les routeurs. Il permet d'avertir un hôte (et uniquement un hôte) d'un meilleur next-hop pour une certaine adresse. Ainsi la passerelle par défaut utilisée (à tort) ne le sera plus ce qui lui permettra d'avoir plus de temps pour traiter d'autres paquets. Ces types sont donc uniquement envoyés par les routeurs à une adresse unicast d'un hôte et ils contiennent l'adresse IP de destination, le next-hop pour atteindre cette adresse et en option l'adresse de couche 2 de la destination. En statefull (cf. DHCPv6 ci-dessous), l'hôte n'est pas au courant de la taille de son préfix, c'est pourquoi il va faire transiter chaque paquet qu'il veut envoyer à une adresse globale par le routeur. Mais si la destination se trouve dans le même sous-réseau, il est absurde de passer par le routeur et c'est pourquoi ce dernier enverra cet ICMPv6. Il faut encore noter que ce message est une information pour le prochain paquet à envoyer à cette même destination. Cela implique que le routeur transmettra tout de même le paquet qu'il a reçu.

La Figure 18 montre comment ces paquets sont construits. On y voit l'adresse de la « Target » et celle par où il faut passer pour l'atteindre : « Destination » (ici directement). Etant donné qu'elle est dans le même sous-réseau, le routeur va aussi donner, dans une option, l'adresse de couche 2 qu'il connait car il a dû lui transmettre le paquet. Finalement il renvoie, comme information, le paquet qu'il a retransmis.

```
⊞ Ethernet II, Src: Cisco_bc:ae:62 (b4:a4:e3:bc:ae:62), Dst: Apple_a9:ba:04 (00:25:4b:a9:ba:04)
⊞ Internet Protocol Version 6, Src: fe80::b6a4:e3ff:febc:ae62 (fe80::b6a4:e3ff:febc:ae62), Dst: 2001:4da0:c00:1:
⊟ Internet Control Message Protocol v6
    Type: 137 (Redirect)
    Code: 0
    Checksum: 0xed99 [correct]
    Reserved: 0 (Should always be zero)
    Target: 2001:4da0:c00:1:f95b:77d9:d445:30dd (2001:4da0:c00:1:f95b:77d9:d445:30dd)
    Destination: 2001:4da0:c00:1:f95b:77d9:d445:30dd (2001:4da0:c00:1:f95b:77d9:d445:30dd)
  ⊟ ICMPv6 Option (Target link-layer address)
      Type: Target link-layer address (2)
      Length: 8
      Link-layer address: c8:0a:a9:2a:ec:0f
  ⊟ ICMPv6 Option (Redirected header)
      Type: Redirected header (4)
      Length: 88
      Reserved: 0 (correct)
      Redirected packet
    ⊞ Internet Protocol Version 6, Src: 2001:4da0:c00:1:62c9:932d:ef29:60e (2001:4da0:c00:1:62c9:932d:ef29:60e),
    ⊞ Internet Control Message Protocol v6
```

Figure 18 : ICMPv6 Redirect

6.6 Gestion du MTU

Contrairement à IPv4, les routeurs ne fragmentent plus les paquets IPv6 qui ont une taille trop grande pour le lien sur lequel ils doivent les transmettre. Par contre, ils renvoient un ICMPv6 qui indique que le paquet n'a pas pu être transmit et qui donne aussi la taille maximale (MTU) qui peut passer sur le lien (au minimum 1280 octets). La Figure 19 nous montre un de ces paquets et on remarque qu'il ressemble à ceux reçu en IPv4 quand la fragmentation était interdite (option activable lors de l'envoi de trames) comme sur la Figure 20.

```
Internet Control Message Protocol v6
  Type: 2 (Too big)
  Code: 0 (Unknown)
  Checksum: 0x50fd [correct]
  MTU: 1300
```

Figure 19 : ICMPv6 Too big

```
Internet Control Message Protocol
  Type: 3 (Destination unreachable)
  Code: 4 (Fragmentation needed)
  Checksum: 0xaf20 [correct]
  MTU of next hop: 1427
```

Figure 20 : ICMP Fragmentation needed

L'avantage de cette façon de faire est que le routeur ne doit plus s'occuper de la fragmentation ce qui laisse plus de performances pour son travail principal qui est le routage. Le désavantage est que l'on peut se trouver dans des situations où un hôte doit envoyer plusieurs fois le même paquet à la suite car il est à chaque fois trop grand pour les routeurs se trouvant sur son chemin. La Figure 21 nous montre un exemple et on voit bien qu'avant l'arrivée du paquet à la destination, il faut l'envoyer trois fois. Par contre, la carte réseau va ensuite se souvenir pendant un certain temps du MTU maximal pour cette destination et va donc directement fragmenter le paquet à la bonne taille. On le remarque car il n'y a plus de « Too big » une fois que le premier paquet a pu atteindre sa destination. Dans un cas pareil avec IPv4, les deux routeurs sur le chemin devraient fragmenter chaque paquet envoyé tandis que là, il y a de la perte de performance qu'une seule fois au début. Comparé à ce qui se passe en IPv4, ce désavantage se transforme même en avantage. C'est encore

un petit détail qui montre qu'IPv6 est clairement là pour améliorer IPv4 et pas uniquement pour donner plus d'adresses IP.

1.391983	2001:4da0:c02:100:ccf	2001:4da0:c02:200:ac90:75	ICMPv6	Echo (ping) request id=0x0001, seq=15
1.402485	2001:4da0:c02:100:ae	2001:4da0:c02:100:ccfb:aa	ICMPv6	Too big (Unknown (0x00))
2.404019	2001:4da0:c02:100:ccf	2001:4da0:c02:200:ac90:75	ICMPv6	Echo (ping) request id=0x0001, seq=16
2.408162	2001:4da0:c02::2	2001:4da0:c02:100:ccfb:aa	ICMPv6	Too big (Unknown (0x00))
3.408057	2001:4da0:c02:100:ccf	2001:4da0:c02:200:ac90:75	ICMPv6	Echo (ping) request id=0x0001, seq=17
3.666621	2001:4da0:c02:200:ac	2001:4da0:c02:100:ccfb:aa	ICMPv6	Echo (ping) reply id=0x0001, seq=17
4.410040	2001:4da0:c02:100:ccf	2001:4da0:c02:200:ac90:75	ICMPv6	Echo (ping) request id=0x0001, seq=18
4.414771	2001:4da0:c02:200:ac	2001:4da0:c02:100:ccfb:aa	ICMPv6	Echo (ping) reply id=0x0001, seq=18

Figure 21 : Envoi multiple du paquet à cause du MTU

7 DHCPv6

L'une des grandes améliorations d'IPv6 est l'attribution des adresses IP. Il n'est, avec IPv6, plus obligatoire d'avoir un serveur DHCP (Dynamic Host Configuration Protocol) comme c'était le cas avant. Il existe aussi plusieurs modes pour l'attribution des adresses et des options et, comme on le voit sur la Figure 22, ce sont les routeurs qui vont indiquer dans leurs paquets « Router advertisement » quelles informations l'hôte doit aller chercher sur le DHCP. Les deux premiers bits de « Flags » sont importants comme on le verra dans les différents modes ci-dessous.

```
⊟ Internet Control Message Protocol v6
    Type: 134 (Router advertisement)
    Code: 0
    Checksum: 0x157a [correct]
    Cur hop limit: 64
  ⊟ Flags: 0x00
      0... .... = Not managed
      .0.. .... = Not other
      ..0. .... = Not Home Agent
      ...0 0... = Router preference: Medium
      .... .0.. = Not Proxied
    Router lifetime: 1800
    Reachable time: 0
    Retrans timer: 0
```

Figure 22 : Options DHCPv6 dans un Router Advertisement

Le choix du mode à utiliser sur un réseau est très compliqué. Il ne suffit pas de savoir comment cela fonctionne, mais il faut voir toutes les implications. Même si avec IPv6 le but est de ne plus avoir besoin de travailler avec les adresses IP, mais uniquement des noms, il est encore fort possible qu'elles soient utilisées dans plusieurs situations et principalement pour la sécurité. Dans ce cas, l'utilisation du DHCP comme en IPv4 peut avoir sa raison. Mais si on n'a pas besoin de connaître et de gérer les adresses IP, autant utiliser un mode plus simple comme le stateless que l'on verra après. Ce travail se concentrant sur la partie réseau d'IPv6, les détails du serveur DHCP ne seront pas abordés, mais ils le sont dans le travail de M. Lienhard.

7.1 Auto-configuration sans DHCP

Comme expliqué précédemment, il n'y a plus forcément besoin de serveur DHCP avec IPv6. Grâce au « Router advertisement » (cf. 6.4), un hôte peut connaître tout ce dont il a besoin du réseau. Avec le préfix reçu, il peut se créer son adresse et contrôler que personne n'a la même sur le lien. Avec le « Router advertisement » il connait aussi l'adresse du routeur et donc de la passerelle par défaut qu'il devra utiliser et il pourra même recevoir l'adresse d'un serveur DNS. Avec cela il a tout pour joindre n'importe quel hôte sur Internet. Cette configuration n'est par contre pas assez complète pour une entreprise qui doit pouvoir avoir un certain contrôle de ce qui se passe sur son réseau. La Figure 22, ci-dessus, implique une auto-configuration sans DHCP car autant le premier bit du « Flag » que le second sont à '0'. Pour que cela fonctionne il est important que le routeur envoie bien le préfix dans ses « Router advertisement » autrement, l'hôte n'aura que son adresse locale de lien.

7.2 Stateless

Si l'on parle de DHCP stateless, cela implique que l'on utilise un serveur DHCP, mais pas totalement comme on le connait avec IPv4. Dans le cas d'une configuration stateless, le serveur ne sert qu'à envoyer des options aux clients. Par contre, l'assignement des adresses aux hôtes se fait toujours en auto-configuration comme vu ci-dessus. Cela permet de donner des indications supplémentaires aux clients sans vouloir avoir un contrôle dessus. C'est le mode qui semble être le plus simple à gérer car il ne faut pas s'occuper de configurer chaque scope du DHCP, mais uniquement ceux qui doivent envoyer des options particulières. Dans ce cas, le routeur va devoir envoyer le flag « Other » dans ses « Router advertisement » en plus du préfix comme nous le montre la Figure 23.

```
⊟ Flags: 0x40
    0... .... = Not managed
    .1.. .... = Other
    ..0. .... = Not Home Agent
    ...0 0... = Router preference: Medium
    .... .0.. = Not Proxied
```

Figure 23 : Flags du mode DHCPv6 stateless

7.3 Statefull

Le mode statefull est celui que l'on connait en IPv4. Cela veut dire que le client ne se construit pas lui-même l'adresse en fonction d'un préfix reçu, mais va la demander au serveur DHCP de même que les options. C'est donc au DHCP de gérer la distribution des adresses ce qui lui permet, par exemple, de faire du logging ou de forcer l'inscription dans le DNS. Par contre, avec DHCPv6, la réservation d'adresses ne se fait plus uniquement en fonction de l'adresse MAC de la carte, mais du DUID (Demand Unique Identifier), une fonction qui prend en compte l'heure de la première requête DHCP et l'adresse MAC. Cela complique la réservation et renforce l'idée que l'on doit tout gérer avec le nom de la machine quelle que soit son adresse IP.

Pour que cela fonctionne, le routeur doit envoyer le flag « Managed » comme nous le montre la Figure 24. Si le routeur envoie un préfix, le même ou non, l'hôte voudra se créer une autre adresse et il en aura donc au minimum deux valides, en même temps, en plus de celle de liaison-locale sur la même interface ce qui n'est absolument pas le but. Il faut donc désactiver l'annonce de préfix sur le routeur. Il existe une option dans les annonces « Router advertisement » qui permet d'envoyer le préfix sans que l'hôte se construise une adresse avec. Cette option n'est, par contre, pas activable sur nos routeurs.

```
⊟ Flags: 0x80
    1... .... = Managed
    .0.. .... = Not other
    ..0. .... = Not Home Agent
    ...0 0... = Router preference: Medium
    .... .0.. = Not Proxied
```

Figure 24 : Flags du monde DHCPv6 statefull

En mode statefull, le serveur DHCP envoie l'adresse sans la longueur du préfix. Un hôte ne sait donc pas dans quel sous-réseau il se trouve étant donné qu'il ne connait que son adresse malgré le fait que tous les préfixes doivent être de longueur /64. Donc, lorsqu'il désire envoyer un paquet à un

hôte du même sous-réseau, il va devoir passer par sa passerelle par défaut (son routeur). Ce dernier va donc transmettre le paquet au destinataire, mais remarquant qu'il doit le renvoyer sur le même lien que celui par où il est venu, il va encore envoyer à l'émetteur le paquet IPCMv6 « Redirect » (cf. 6.5) afin qu'ils ne passent plus par lui pour s'envoyer des paquets entre eux.

7.4 Exemple d'utilisation des modes

On peut s'imaginer utiliser ces trois façons de récupérer des adresses et des options :

7.4.1 Auto-configuration sans DHCP

Dans une entreprise, on peut activer ce mode pour le réseau des imprimantes dans lequel on met aussi une interface du serveur d'impression. Etant donné que les imprimantes ne doivent pas forcement recevoir des options et que leur adresse IP ne doit pas être connue, il nous importe peu de les avoir dans le DHCP et dans le DNS.

Ce mode est bien-entendu plus utilisé pour les configurations dites privées, lorsque l'on n'a qu'un réseau simple derrière un routeur comme c'est le cas avec l'ADSL à la maison.

7.4.2 Stateless

On peut très bien s'imaginer une configuration stateless sur le réseau qui contient les téléphones IP. Ici aussi, leur adresse IP ne doit pas être connue sur le DHCP ou le DNS. Par contre il faut absolument leur transmettre des options afin qu'ils sachent où se trouve, par exemple, leur « Registrar », serveur d'enregistrement.

7.4.3 Statefull

Une entreprise ou une école peut mettre tous les réseaux des ordinateurs utilisateurs en mode DHCP statefull car elles désirent pouvoir retrouver quel ordinateur (adresse MAC) avait quelle adresse IP à un moment précis. Comme on l'a vu précédemment, en étant en statefull il est facile de garder des traces. On utilisera donc ce mode principalement pour des raisons de sécurité.

7.5 Fonction ip helper / dhcp relay

Lorsque le client et le serveur DHCP qu'il essaie de joindre sont séparés par un routeur (ou plusieurs), il faut que ce dernier transmette les paquets au serveur en encapsulant la trame envoyée par le client dans un paquet unicast. Cela est dû au fait que les trames envoyées à un serveur DHCP, donc via une adresse multicast, ne passent pas les routeurs. En IPv4, la commande « ip helper » permet de transmettre un grand nombre de protocoles (TFTP, DNS, Time, NetBIOS, ND ou DHCP) à une adresse IP. Même si l'on ne voulait transmettre que les paquets DHCP, les autres protocoles étaient aussi transmis si l'on ne triait pas manuellement. En IPv6, une commande ne s'occupant que du DHCP a été créée : « ipv6 dhcp relay destination <adresseIPv6> » et elle va donc transmettre les paquets envoyés par un hôte à l'adresse multicast DHCP directement et en unicast à l'adresse que l'on a donnée. Cette commande n'aura d'influence que pour les paquets en direction du serveur DHCP. Il faudra donc l'activer sur tous les sous-réseaux qui seront configurés en DHCP statefull ou stateless sauf celui où se trouve le serveur DHCP, car autrement, il recevrait à double tous les paquets : Une fois directement depuis le client et une deuxième fois encapsulés par le routeur.

La Figure 25 nous montre le paquet relayé qui est arrivé au serveur DHCPv6. On peut remarquer que l'adresse de destination est la globale du serveur et que celle de source est l'interface par laquelle le

paquet a été envoyé depuis le routeur sur lequel le poste client s'est connecté. Pour que le serveur sache dans quel sous-réseau se trouve ce client, le routeur donne son adresse globale qui est sur le même lien que le client (« Link address »). C'est dans « Relay Message » que l'on va retrouver le paquet envoyé par le client. Le serveur répondra de la même manière, à l'adresse source du paquet qu'il a reçu et le routeur désencapsulera et transmettra au client qui avait fait la requête. Ceci est donc totalement transparent pour le client.

```
⊞ Ethernet II, Src: Cisco_6d:7e:43 (00:26:98:6d:7e:43), Dst: Vmware_a3:00:77 (00:50:56:a3:00:77)
⊞ Internet Protocol Version 6, Src: 2001:4da0:c02::e (2001:4da0:c02::e), Dst: 2001:4da0:c01:0:d939:58f5:1798:5
⊞ User Datagram Protocol, Src Port: dhcpv6-server (547), Dst Port: dhcpv6-server (547)
⊟ DHCPv6
    Message type: Relay-forw (12)
    Hopcount: 0
    Link address: 2001:4da0:c00:1:b6a4:e3ff:febc:ae62 (2001:4da0:c00:1:b6a4:e3ff:febc:ae62)
    Peer address: fe80::ac90:758c:83fc:4c9c (fe80::ac90:758c:83fc:4c9c)
  ⊟ Relay Message
      Option: Relay Message (9)
      Length: 89
      Value: 010ce5d2000800020c1c0001000e000100011549557800025...
    ⊞ DHCPv6
  ⊞ Interface-Id
```

Figure 25 : Paquet DHCPv6 relayé

Dans un grand réseau et sous certaines conditions, on pourrait utiliser la même adresse IP pour tous les serveurs DHCP et grâce au routage, le plus proche (ou le meilleur) sera atteint. Cela peut faciliter les configurations du réseau mais ne peut, par contre, pas fonctionner en statefull. Dans ce cas, on appellera l'adresse du serveur « anycast ».

7.6 Commandes

7.6.1 Auto-config sans DHCP

Pour ce mode, les seules commandes entrées au point 5.4 mettent déjà l'interface en mode auto-config sans DHCP.

7.6.2 Stateless

conf t	
interface <interface>	Seulement sur les interfaces qui implémentent déjà IPv6
ipv6 nd other-config-flag	Envoie le flag "Other" dans les "Router Advertisement"
end	

7.6.3 Statefull

conf t	
interface <interface>	Seulement sur les interfaces qui implémentent déjà IPv6

ipv6 nd prefix default no-advertise	Pour ne pas envoyer le prefix dans les "Router Advertisement"
ipv6 nd managed-config-flag	Envoie le flag "Managed" dans les "Router Advertisement"
end	

7.6.4 DHCP relay

conf t	
interface <interface>	Seulement sur les interfaces qui implémentent déjà IPv6
ipv6 dhcp relay destination <adresse>	Transmet toutes les requêtes DHCPv6 à l'adresse IPv6 entrée dans une trame unicast. Peut être mise plusieurs fois.
end	

8 Les VRF

Quand on sait que le SIEN s'occupe des réseaux de la police, des hôpitaux, des écoles publiques et de l'administration, on peut facilement s'imaginer qu'il serait mauvais que ces derniers puissent être mélangés entre eux. Pour éviter cela, il y a deux façons de faire : La première est de séparer physiquement les quatre réseaux, solution très couteuse car ils se partagent, à plusieurs endroits, des bâtiments et il faudrait donc acheter tous les équipements à triple ou quadruple. L'autre solution est de diviser le réseau physique en plusieurs réseaux logiques qui ne doivent pas se connaître et donc pas pouvoir communiquer entre eux sans passer par un firewall. C'est la technologie des VRF (Virtual Routing and Forwarding) qui permet de se partager un équipement en ayant plusieurs tables de routage totalement indépendantes. Chaque réseau logique est donc appelé VRF. Chaque interface appartient à une VRF et ne peut donc pas transmettre les paquets d'une autre VRF.

Jusqu'à présent, les VRF ont été configurées avec la commande « ip vrf <VRFName>» ce qui implique qu'elles n'étaient appliquées qu'au protocole IPv4. Cisco a ajouté une nouvelle commande, « vrf definition <VRFName> », qui va permettre d'y intégrer autant les réseaux IPv4 qu'IPv6. Après avoir créé une VRF, il faut encore activer soit IPv4, soit IPv6, soit les deux avec les commandes « adress-family ipv4 » et « adress-family ipv6 » sans quoi on trouve des messages d'erreurs comme sur la Figure 26. Cette nouvelle définition de VRF permet de mettre des « route target » séparées pour les réseaux IPv4 et IPv6 si on le désire.

```
3750GAR1(config-if)#ip add 18.22.33.11 255.255.255.0
%Loopback99 is linked to a VRF. Enable IPv4 on that VRF first.
3750GAR1(config-if)#ipv6 add 2001:4da0::3/64
%Loopback99 is linked to a VRF. Enable IPv6 on that VRF first.
```

Figure 26 : Message d'erreur lorsque les protocoles ne sont pas activés dans les VRF

Pour les appareils où se trouvent déjà des VRF uniquement IPv4, Cisco a créé une commande, « vrf upgrade-cli multi-af-mode common-policies », qui va convertir les anciennes définitions en de nouvelles auxquelles on pourra donc aussi faire correspondre un réseau IPv6. Cette commande est très utile car elle n'interrompt et ne modifie pas les services en cours. Avec un argument supplémentaire, on peut aussi l'appliquer à une seule VRF.

8.1 Compatibilités

La nouvelle définition des VRF est compatible avec tout le matériel à disposition. Par contre, il n'est pas encore possible d'activer IPv6 dans les VRF des petits switchs L2-L3 (3560 et 3750G). La Figure 27 nous montre le message d'erreur indiqué et au vu de celui-ci, il y a de fortes chances qu'il ne sera jamais possible de l'activer malgré le fait que la commande y soit bien présente.

```
3750G(config-vrf)#address-family ipv6
 IPv6 VRF not supported for this platform or this template
```

Figure 27 : Message d'erreur lors de l'activation d'IPv6 dans une VRF sur un switch 3750G

Pour les autres équipements, les compatibilités IPv6 avec les VRF seront analysées plus précisément dans les différents chapitres concernés mais le Tableau 2 en constitue un résumé.

	3560	3750G	6509	1841	1921	2811	7206
VRF IPv6	-	-	X	X	X	X	X
Interfaces	-	-	X	X	X	X	X
Routage statique	-	-	X	X	X	X	X
OSPFv3	-	-	-	-	-	-	-
EIGRP	-	-	-	-	-	-	-
BGP / VPN	-	-	X	?(1)	?(1)	?(1)	X
Multicast	-	-	-	X	-	X	X
(1) : Pas testé car ces routeurs n'implémentent pas BGP au SIEN							

Tableau 2 : Compatibilités des VRF IPv6 sur les équipements du SIEN

8.2 Gestion de la mémoire

Avoir des VRF implique avoir plusieurs tables de routage distinctes. Comme on le verra en analysant la mémoire d'un routeur (cf. Performances, 12.1), ce n'est pas la création d'une VRF qui prend de la place mémoire, mais l'activation d'un protocole dans celle-ci car le routeur doit en réserver pour la table de routage. Mais la place prise n'est pas très grande comparé à celle que prend une instance d'OSPF. Cela veut dire que les VRF ne sont pas gourmandes en mémoire et que l'on peut en utiliser un grand nombre sans devoir faire attention. Par contre, il faudra faire attention à la place générale prise par les tables de routage lorsque le routeur passera en dual-stack. C'est bien ceci qui pourra poser des problèmes, mais pas le nombre de VRF.

8.3 Commandes

conf t	
mls ipv6 vrf	**6509 uniquement** : obligatoire pour créer des address-family IPv6
vrf upgrade-cli multi-af-mode common-policies [vrf <VRFName>]	Transforme les anciennes définitions de VRF en de nouvelles. (Peut se faire pour une seule VRF)
vrf definition <VRFName>	Entre dans le mode de configuration d'une VRF.
rd <rdNB1:rdNB2>	Définit un « route distinguisher »
route-target {import ; export ; both} <rtNB1:rtNB2>	Définit une route-target pour toutes les address-family
address-family ipv4	Crée et rentre dans la configuration IPv4 de la VRF
route-target {import ; export ; both} <rtNB1:rtNB2>	Définit une route-target pour les adresses IPv4
address-family ipv6	Crée et rentre dans la configuration IPv6 de la VRF
route-target {import ; export ; both} <rtNB1:rtNB2>	Définit une route-target pour les adresses IPv6

end	
show vrf	Montre toutes les VRF existantes, les protocoles qu'elles gèrent et les interfaces sur lesquelles elles sont actives

9 OSPFv3

Le protocole de routage utilisé au SIEN est OSPF (Open Shortest Path First) et c'est donc celui-ci qui est analysé. Une nouvelle version d'OSPF a été créée pour prendre en charge IPv6, il s'agit d'OSPFv3. Dans le fonctionnement, il n'y a pas de différence entre cette nouvelle version et l'ancienne, mais quelques améliorations ont tout-de-même été apportées. On ne parle, dans cette version, plus de sous-réseaux, mais de préfixes et on ne parle donc plus de masques, mais de longueurs du préfix. Les interfaces ne doivent plus non-plus avoir une adresse IP configurée pour transmettre les paquets OSPF, car elles utilisent uniquement les adresses de liaison-locales (cf. 3.2.1). Finalement, les adresses multicast sont devenues « ff02::5 » pour joindre tous les routeurs et « ff02::6 » pour joindre le DR (Designated Router) et le BDR (Backup Designated Router). On peut encore noter que les routeurs doivent toujours avoir un ID de 32 bits donc, si l'on n'a pas d'interface en IPv4 il faudra donner expressément un ID à OSPF. De plus, l'authentification entre les routeurs n'est plus interne à OSPF mais utilise directement les options d'IPv6.

9.1 Multi-area

Etant donné qu'OSPFv3 n'a pas fondamentalement changé par rapport à OSFPv2, les réseaux implémentant ce protocole de routage sont toujours construits en différentes zones appelées « area ». Ces dernières sont généralement distribuées géographiquement et sont toutes reliées à l' « area 0 » que l'on nomme aussi « backbone ». Cette disposition est utilisée au SIEN et va donc aussi être utilisée sur le deuxième prototype alors que le premier nous a permis de comprendre la base d'OSPF.

Comme vu sur le plan d'adressage au point 4.2.2, le réseau de test est composé de quatre areas en plus du backbone. Une des areas est « stub » ce qui implique qu'elle ne connaît que les routes internes au réseau OSPF et donc pas celles redistribuées à l'intérieur depuis un autre protocole de routage ou une autre instance d'OSPF. Cela implique qu'une route par défaut doit être connue afin de pouvoir joindre toutes les autres adresses. Deux autres areas sont des « nssa » ce qui veut dire « Not-So-Stubby Area » mais qui sont plus précisément des « Totally Not-So-Stubby Area », car elles ne connaissent que leurs propres routes (internes à l'area) mais permettent tout-de-même d'en importer depuis l'extérieur et de les transmettre au backbone. Cette fonction est souvent utilisée car on ne sait jamais si l'on devra un jour importer des routes de l'extérieur. Une dernière area est standard et connait toutes les routes de tous les réseaux. C'est elle qui sera ensuite connectée à Internet ce qui implique qu'elle sera la route par défaut pour le backbone afin qu'il ne doive pas connaître toutes les routes d'Internet (ce qui serait beaucoup trop lourd pour le matériel que l'on a à disposition et totalement inutile). Finalement, le backbone est lui-aussi une area normale car il doit absolument connaître toutes les routes du réseau (sauf celles d'internet) afin de pouvoir router les paquets vers les bonnes areas.

Comme déjà vu, IPv6 a aussi été fait en pensant aux tables de routage avec comme but de les simplifier au maximum. Afin d'éviter de transmettre dans le backbone tous les différents préfixes des sous-réseaux d'une zone, il est possible de faire de l'agrégation de réseaux. Cela veut dire que l'on va regrouper plusieurs sous-réseaux avant de les transmettre grâce à OSPFv3. La Figure 28 montre le retour de la commande « show ipv6 route » avant et après avoir activé l'agrégation sur un autre routeur. L'agrégation s'active sur les routeurs ABR (Area Border Router) dans la configuration d'OSPFv3 grâce à la commande « area <areaNo> range <IPv6Address/PrefixLenght> ». Sur la Figure

28 la commande entrée est donc : « area 2 range 2001 :4da0 :c01 :200 ::/56 ». Si dans les routes agrégées se trouvent plusieurs coûts différents, le plus élevé est pris pour la route transmise. Cela ne pose pas de problèmes vu la configuration de notre réseau (et celui du SIEN), mais dans le cas où un sous-réseau possède plusieurs sorties en direction des autres réseaux, cela pourrait faire prendre des routes moins optimales pour certains paquets. C'est donc dans ce cas qu'il faut faire attention lorsque l'on fait des agrégations de réseaux.

```
7206PE#sh ipv6 route
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       D - EIGRP, EX - EIGRP external, ND - Neighbor Discovery, l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OI   2001:4DA0:C01:200:219:AAFF:FE00:AB18/128 [110/1]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.7
OI   2001:4DA0:C01:201:219:AAFF:FE00:AB18/128 [110/1]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.7
OI   2001:4DA0:C01:202:219:AAFF:FE00:AB18/128 [110/1]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.7
OI   2001:4DA0:C01:2A0:219:AAFF:FE00:AB18/128 [110/1]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.7
OI   2001:4DA0:C01:2A1:219:AAFF:FE00:AB18/128 [110/1]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.7
OI   2001:4DA0:C01:2A2:219:AAFF:FE00:AB18/128 [110/1]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.7
L    FF00::/8 [0/0]
     via Null0, receive
7206PE#sh ipv6 route
IPv6 Routing Table - default - 2 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       D - EIGRP, EX - EIGRP external, ND - Neighbor Discovery, l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OI   2001:4DA0:C01:200::/56 [110/1]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.7
L    FF00::/8 [0/0]
     via Null0, receive
```

Figure 28 : Exemple d'agrégation de réseaux

Sachant que l'on peut faire 256 sous-réseaux avec un préfix de longueur /56, on voit que l'on peut rapidement réduire la taille des tables de routage du backbone et des areas normales et « stub ».

Le routage purement OSPF mais multi-ara a été testé et fonctionne parfaitement avec OSPFv3. Il n'y a aucune différence avec OSPFv2 excepté les commandes comme il sera expliqué ci-dessous.

9.2 Multi-VRF

S'il n'y avait aucun problème d'implémentation de ces VRF avec OSPFv2, ce n'est pas le cas avec OSPFv3. Pour l'instant, Cisco n'a pas encore sorti l'IOS intégrant cette fonctionnalité pour notre matériel et il faut dire que nos routeurs ne sont pas encore OSPFv3 VRF-aware (terme officiel pour indiquer qu'il supporte ces dernières). Lorsque l'on désire activer OSPFv3 sur une interface appartenant à un VRF, le message d'erreur indiqué à la Figure 29 apparait. On sait donc, dès ce moment, que l'on ne va pas pouvoir créer notre réseau IPv6 comme on le désire mais, pour pouvoir

tout de même tester certaines autres fonctionnalités, il est possible d'utiliser les VRF avec un routage statique et c'est ce qui est fait à une plus petite échelle que ce qui avait été prévu. La Figure 6, vue précédemment nous montre ce qui a été finalement effectué.

```
6509PE(config-subif)#ipv ospf 112 ar 4
% OSPFv3 not supported on VRF interface
```

Figure 29 : Message d'erreur lors de l'activation d'OSPFv3 sur une interface VRF

9.3 Fonctionnement

Premièrement, il faut activer le routing IPv6 que ce soit sur les routeurs ou sur les switchs L3. Cela se fait grâce à la commande « ipv6 unicast-routing ». Ensuite, l'implémentation d'OSPF pour IPv6 dans les routeurs et switchs L3 est très semblable à ce que l'on fait pour IPv4. Il suffit de remplacer « ip » par « ipv6 » ou de le préfixer. Pour activer OSPFv3 on tapera donc en mode de configuration global « ipv6 router ospf <instance No> » alors que pour OSPFv2 c'est « router ospf <instance No> ».

Ensuite il faut aussi l'activer sur les interfaces afin que les différents sous-réseaux soient transmis par celles-ci. Pour cela on utilise la commande « ipv6 ospf <instance No> area <area No> ». A partir de là, tous les sous-réseaux connectés aux interfaces avec OSPF activé vont être envoyés sur OSPF. Il est très important de noter que l'ajout de réseaux avec la commande « network <réseau> <masque inversé> » n'est plus possible avec OSPFv6. Cela est dû au fait que l'on ne doit plus avoir d'adresse IP connue entre deux routeurs mais que l'on peut s'appuyer uniquement sur les adresses de liaison-locale. Ainsi OSPFv6 ne s'active plus que directement sur les interfaces comme on le voit ci-dessous.

9.4 Commandes

conf t	
ipv6 unicast-routing	Active le routage IPv6
ipv6 router ospf <process-id>	Active OSPFv3 sur le routeur/switch L3
router-id <id>	Pour donner une identification au routeur OSPF. Doit être mis en format IPv4.
area <area-id> {stub ; nssa [no-summary]}	Pour donne un type à une area
area range <IPv6Address/prefixLength>	Se met sur les ABR pour transmettre le préfix donné à la place de plusieurs qu'il contient. Prend le plus grand métrique.
default-information originate [always]	Envoie une route par défaut (::/0)
interface <interface>	Seulement sur les interfaces qui implémentent déjà IPv6
ipv6 ospf <process-id> area <area-id>	Active OSPFv3 sur l'interface
end	

show ipv6 ospf	Pour voir les détails d'OSPFv3
show ipv6 route ospf	Pour voir les routes apprises par OSPFv3

10 Backbone BGP/MPLS-VPN

Comme vu ci-dessus, la partie centrale du réseau est appelée « backbone ». Dans le réseau du SIEN, ce dernier est très important car il relie tous les sites répartis sur environ 800km^2[3] entre eux et, est donc composé d'un grand nombre de routeurs. Il serait possible, dans ce backbone, d'avoir une instance OSPF pour chaque VRF, mais cela impliquerait beaucoup de tables de routage et une perte de performances, car dans ce cas, chaque paquet doit être analysé jusqu'à la couche trois pour savoir par où il doit être envoyé.

Afin d'améliorer les performances d'un backbone, il existe le protocole MPLS (MultiProtocol Label Switching) qui va, lui, taguer les paquets à l'entrée afin que tous les routeurs du backbone regardent uniquement ce tag. On évite ainsi de passer dans toute la table de routage étant donné que ce tag est mis avant la couche IP. Une fois la trame arrivée au routeur de destination du backbone, ce dernier va la transmettre en fonction de la table de routage au bon routeur qui va se trouver dans une autre area OSPF. Pour ces routeurs, qui sont dans les différentes areas, le routage interne au backbone n'est pas connu et n'a donc aucune influence. C'est pourquoi ce type de backbone est souvent utilisé chez les fournisseurs d'accès qui doivent pouvoir relier le réseau d'une entreprise entre plusieurs sites de manière sécurisée et transparente pour le client.

A l'intérieur du backbone, il faut tout-de-même pouvoir s'échanger les différentes routes. Il faut donc, là aussi, un protocole de routage et celui qui sait gérer plusieurs VRF sur un seul lien est BGP. Il va, lui aussi, mettre des labels afin de ne pas mélanger les différentes VRF. Ces labels seront gardés tout au long du passage dans le backbone. A ce moment, on dit qu'un VPN (Virtual Private Network) est monté. C'est donc pour cette raison qu'on appelle notre backbone « BGP/MPLS-VPN ».

10.1 BGP

Le protocole de routage BGP se diffère des autres car il fonctionne sur TCP. Cela implique qu'il n'est pas dépendant de la version d'IP utilisée. Il faut connecter BGP à ses « neighbor » (voisins) et ensuite lui indiquer les routes qu'il doit transmettre. Ainsi, on peut se connecter à ses neighbors en IPv4 et leur transmettre les routes en IPv6 ou inversement. Dans notre cas, on utilisera iBGP. Le « i » indique que l'on a à faire à un « intra Autonomous System », ce qui est l'équivalent d'une area OSPF mais à l'échelle d'internet. Dans la zone où on veut utiliser BGP, il faut tout-de-même avoir un IGP (Interior Gateway Protocol) comme OSPF qui sert uniquement à ce que tous les routeurs du backbone se connaissent entre eux. Une fois que l'IGP est actif, on peut les interconnecter et faire monter ce qui s'appelle un « peering ». Dès cet instant, BGP va s'occuper de transmettre ses routes à tous ses neighbors.

Pour résumé, BGP n'est absolument pas influencé par IPv6 et est capable de transporter autant des routes IPv4 qu'IPv6 à des peers IPv4 ou IPv6. De ce fait, les commandes ne changent pas fondamentalement et sont toujours entrées dans « router bgp <ASno> ». On travaille aussi avec des « address-family » et il suffit de remplacer, logiquement, « ipv4 » par « ipv6 ». On aura donc des « address-family ipv6 <VRFName> » qui s'occuperont d'introduire des routes de VRF IPv6 dans BGP, soit avec la commande « network », soit avec des « redistribute » et on aura une « address-family vpnv6 » qui s'occupera, elle, de faire la connexion et d'envoyer les routes IPv6 à ses peers. On y trouvera donc exactement les mêmes commandes que dans « address-family vpnv4 ». La limitation des peers IPv6 se fera à cause de MPLS comme on le verra ci-dessous et c'est donc la raison pour

laquelle on ne monte pas notre réseau BGP en dual-stack. La Figure 30 nous montre comment sont apprises et transmises les routes d'une VRF dans le backbone. On voit que le prochain saut (next hop) est une adresse IPv4 transformée en IPv6 et cette dernière est l'adresse de sortie du backbone d'où l'obligation d'avoir un IGP à l'intérieur de celui-ci. Le label affiché est celui lié au VPN et va être uniquement utilisé par le routeur de sortie. La Figure 31 nous montre comment la table de routage reprend ces informations et on voit que toutes les routes apprises par BGP ont un « B » devant.

```
7206PE#show bgp vpnv6 unicast all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65500:2120 (police)
   2001:4DA0:C02::2/128
                    ::                  29/nolabel(police)
   2001:4DA0:C02:100::1/128
                    ::FFFF:192.168.1.1
                                        nolabel/45
   2001:4DA0:C02:200::/64
                    ::                  30/nolabel
   2001:4DA0:C02:A00::/64
                    ::FFFF:192.168.1.1
                                        nolabel/29
   2001:4DA0:C02:A01::/64
                    ::FFFF:192.168.1.1
                                        nolabel/30
   2001:4DA0:C02:AA0::/64
                    ::FFFF:192.168.1.1
                                        nolabel/31
   2001:4DA0:C02:AA1::/64
                    ::FFFF:192.168.1.1
                                        nolabel/32
```

Figure 30 : Adresses inclues dans le VPN (VRF) de « police » avec les labels

```
7206PE#show ipv6 route vrf police
IPv6 Routing Table - police - 8 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       D - EIGRP, EX - EIGRP external, ND - Neighbor Discovery, l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
LC  2001:4DA0:C02::2/128 [0/0]
     via Loopback212, receive
B   2001:4DA0:C02:100::1/128 [200/0]
     via 192.168.1.1%default, indirectly connected
S   2001:4DA0:C02:200::/64 [1/0]
     via FE80::219:AAFF:FE00:AB18, GigabitEthernet0/3.212
B   2001:4DA0:C02:A00::/64 [200/0]
     via 192.168.1.1%default, indirectly connected
B   2001:4DA0:C02:A01::/64 [200/0]
     via 192.168.1.1%default, indirectly connected
B   2001:4DA0:C02:AA0::/64 [200/0]
     via 192.168.1.1%default, indirectly connected
B   2001:4DA0:C02:AA1::/64 [200/0]
     via 192.168.1.1%default, indirectly connected
L   FF00::/8 [0/0]
     via Null0, receive
```

Figure 31 : Table de routage IPv6 d'une VRF

10.2 MPLS

MPLS est la technique du tagage pour faire passer les paquets plus rapidement à travers le réseau du backbone. Pour faire cela, il existe deux protocoles : TDP (Tag Distribution Protocol) et LDP (Label Distribution Protocol) et c'est ce dernier qui est utilisé au SIEN. On voit sur la Figure 34 qu'il y aura deux tags lors de l'entrée du paquet dans le backbone. Le premier sert à trouver le chemin à l'intérieur de celui-ci et est donc lié à MPLS, alors que le second est le tag VPN qui devrait correspondre à celui vu sur la Figure 30. Le nexthop n'a pas d'influence dans ce cas, car le paquet aura uniquement les labels avant l'entête IP. Et c'est grâce au premier label que le prochain routeur saura où transmettre cette trame. On arrive donc à imaginer le tunnel qui se forme et qui permet de gagner en performance en évitant d'aller regarder dans la couche trois des paquets. Et ainsi, les routeurs intermédiaires ne doivent pas connaitre toutes les adresses de tous les réseaux, mais uniquement les tags correspondants aux routeurs PE du backbone.

Si la transition à IPv6 a posé beaucoup de problèmes et a pris du temps, cela vient principalement du fait que MLPS ne supporte pas encore IPv6 pour échanger les tags. On ne trouve malheureusement que très peu de littératures à ce propos, ce qui est étonnant car il y en a un grand nombre sur IPv4. Cela nous laisse conclure que peu de personnes ont dû avoir des résultats probants et n'ont donc pas pu les partager. Il y a, malgré tout, une présentation de Cisco qui le confirme [4]. De plus, le fait qu'aucun protocole IGP n'est VRF-aware (EIGRP a aussi été testé comme le montre la Figure 32) pose déjà une première barrière, car au SIEN, « BGP/MPLS » prend tout son sens avec les réseaux virtuels.

```
6509PE(config-subif)#ipv eigrp 1
%EIGRP: Classic IPv6 command not supported on VRF interface GigabitEthernet
2/23.115
```

Figure 32 : Message d'erreur lors de l'activation d'EIGRP sur une interface VRF

La seule solution pour faire transiter des paquets IPv6 tagués est d'utiliser un backbone en IPv4 grâce à la technique d'intégration 6VPE qui va être présentée ci-dessous.

On peut encore noter que lorsque l'on capture des paquets entre deux routeurs du backbone, on ne voit rien en dessous de la couche trois. On ne peut donc pas voir les tags devant les paquets. Par contre, on remarque qu'il n'y a pas de couche deux dans un backbone MPLS. La Figure 33 en donne un exemple.

```
⊞ Frame 6: 100 bytes on wire (800 bits), 50 bytes captured (400 bits)
⊟ Raw packet data
    No link information available
⊞ Internet Protocol Version 6, Src: 2001:4da0:c01:a000::2 (2001:4da0:c01:a000::2), Dst: 2001:4da0:c01:a001::1 (2001:4da0:c01:a001::1)
⊟ Internet Control Message Protocol v6
    Type: 128 (Echo (ping) request)
    Code: 0 (Should always be zero)
    Checksum: 0x6ee4
    ID: 0x1c7e
    Sequence: 2
  ⊞ Data (2 bytes)
```

Figure 33 : Capture d'un paquet entre deux routeurs du backbone

```
7206PE#show ipv6 cef vrf police
::/0
  no route
::/127
  discard
2001:4DA0:C02::2/128
  receive for Loopback212
2001:4DA0:C02:100::1/128
  nexthop 192.168.2.10 GigabitEthernet0/2 label 23 45
2001:4DA0:C02:200::/64
  nexthop FE80::219:AAFF:FE00:AB18 GigabitEthernet0/3.212
2001:4DA0:C02:A00::/64
  nexthop 192.168.2.10 GigabitEthernet0/2 label 23 29
2001:4DA0:C02:A01::/64
  nexthop 192.168.2.10 GigabitEthernet0/2 label 23 30
2001:4DA0:C02:AA0::/64
  nexthop 192.168.2.10 GigabitEthernet0/2 label 23 31
2001:4DA0:C02:AA1::/64
  nexthop 192.168.2.10 GigabitEthernet0/2 label 23 32
FE80::/10
  receive for Null0
FF00::/8
  multicast
```

Figure 34 : Affichage des tags à l'entrée dans le backbone pour les préfixes IPv6 d'une VRF

10.2.1 Technique d'intégration 6VPE

La technique d'intégration 6VPE nous permet d'avoir un backbone VPN supportant IPv6 sans que tous les équipements de celui-ci doivent implémenter cette nouvelle version d'IP. Le premier avantage de cette technique est que l'on ne doit rien modifier au backbone existant mais seulement ajouter les préfixes que l'on désire router. Cela implique que l'on n'a pas un backbone dual-stack mais que l'on crée une sorte de tunnel grâce aux tags car, comme on l'a vu, le paquet est tagué à l'entrée du backbone et, n'a donc plus besoin d'utiliser d'IP à l'intérieur. Cette technique permet de garder du matériel plus ancien qui pourrait même ne pas supporter IPv6. Par contre, tous les routeurs du bord du backbone (routeurs appelés PE) doivent eux, logiquement, être dual-stack pour pouvoir communiquer avec ceux des clients (appelés CE). Dans le cas particulier du SIEN, étant donné que les routeurs du backbone sont tous des PE, on va uniquement profiter de cette technique pour faire transiter les trames IPv6 mais, par contre, tous les équipements devront tout-de-même être dual-stack. Grâce à cette technique, on évitera, même pour le SIEN, de devoir modifier trop de configurations.

La Figure 35 nous montre un exemple de cette technique d'intégration. Dans le réseau du SIEN il y a beaucoup plus de routeurs « 6VPE » et pas de routeur « P ». Mais le fonctionnement est semblable.

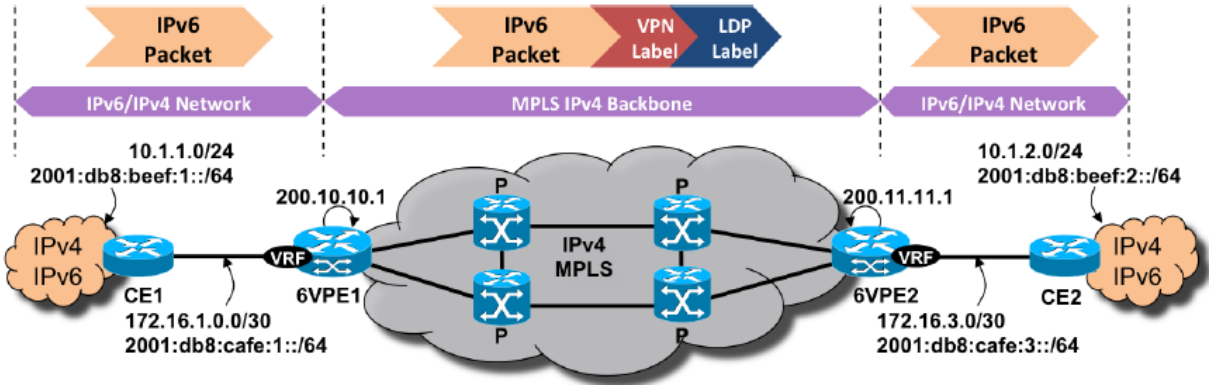

Figure 35 : Illustration de la technique 6VPE (source : http://www.cisco.com/web/FR/documents/pdfs/newsletter/ciscomag/2010/01/ciscomag_30__bn-ipv6.pdf)

Dans le futur, quand IPv6 sera généralisé, on parle déjà de la technique 4PE qui permettra de faire transiter les trames IPv4 sur le backbone IPv6. Ainsi il n'y aura jamais de backbone dual-stack et cela devrait permettre d'avoir des performances maximales.

Cette technique a donc dû être utilisée pour faire fonctionner le backbone étant donné que MPLS n'est pas encore capable de s'échanger les tags sur un réseau IPv6, et elle a fonctionné parfaitement et simplement. On voit, sur la Figure 36, que la première adresse de passage est en IPv4 (transformée en IPv6 pour que le « traceroute » la comprenne). Cette capture a été faite lorsque le lien entre les deux routeurs PE a été coupé pour forcer le passage par un routeur intermédiaire qui n'est absolument pas configuré pour IPv6.

```
7206PE#traceroute vrf police 2001:4DA0:C02:A00::1
Type escape sequence to abort.
Tracing the route to 2001:4DA0:C02:A00::1

  1 ::FFFF:192.168.2.10 [MPLS: Labels 23/29 Exp 0] 0 msec 0 msec 4 msec
  2 2001:4DA0:C02:100::1 [MPLS: Label 29 Exp 0] 0 msec 0 msec 0 msec
  3 2001:4DA0:C02:A00::1 0 msec 4 msec 0 msec
```

Figure 36 : Traceroute d'une adresse IPv6 à travers le backbone

Techniquement, il n'y a rien besoin de faire de spécial pour que 6VPE fonctionne. Il suffit d'envoyer ses routes à des peers IPv4 et d'activer le MPLS, avec la commande « ip mpls » sur les interfaces désirées. Il est recommandé de donner, comme adresse de connexion, celle d'une interface de loopback implémentant IPv4 afin que le routeur soit toujours joignable même si une interface physique venait à tomber.

10.3 Commandes

10.3.1 BGP

conf t	
router bgp <ASno>	Active BGP sur le routeur

bgp router-id <id>	Pour donner une identification au routeur BGP. Doit être mis en format IPv4.
neighbor <IPAddress> remote-as <ASno>	Pour se connecter à un « voisin ». L'adresse peut être en IPv6 ou IPv4, mais pour que cela fonctionne avec MPLS il faut de l'IPv4. L'ASno doit être le même que pour le routeur BGP afin de faire de l'iBGP.
neighbor <IPAddress> update-source <interface>	Pour créer un peering à travers les loopback pour que cela fonctionne même à la perte d'une interface
addres-family ipv6 vrf <VRFName>	Pour définir les routes à envoyer dans BGP pour une VRF.
redistribute {connected ; static ; ospf <…>}	Pour récupérer des préfixes connectés où d'une autre instance de routage
network <IPv6Address/prefixLength>	Ajoute un préfix à BGP. (Pas utilisé au SIEN)
exit-address-family	
address-family vpnv6	Pour entrer dans la configuration de la partie VPN de BGP
neighbor <IPv6Address> activate	Indique que l'on veut envoyer nos routes à ce voisin qui doit être déjà en peering
neighbor <IPv6Address> send-community extended	Envoie les routes avec les rd
end	
show ipv6 route vrf <VRFName>	Pour voir si des routes ont bien été échangées pour une VRF
show bgp vpnv6 unicast all labels	Pour voir si des labels VPN ont bien été échangés.

10.3.2 MPLS

conf t	
ipv6 cef	Activation du protocole propriétaire CEF (Cisco Express Forwarding) de Cisco (obligatoire pour le MPLS).
mpls ldp router-id <interface>	Mettre de préférence une interface loopback pour que le routeur soit joignable de partout. Doit être la même que le « update-source » de BGP.
mpls label protocol ldp	Pour forcer le protocole MPLS à LDP
interface <interface>	Seulement sur les interfaces du backbone
mpls ip	Force l'interface à utiliser les tags MPLS pour l'envoi des paquets de cette interface
end	
show mpls forwardong-table	Montre les différentes liaisons MPLS

11 Multicast

Le multicast est la fonctionnalité qui permet d'envoyer un paquet à plusieurs destinations. S'il n'est pas implémenté sur Internet en IPv4, il sera obligatoire en IPv6. Par contre, il est déjà très utilisé dans les réseaux intra-entreprises quand on peut avoir le contrôle de tous les équipements par lesquels le paquet doit passer. Le multicast est, par exemple, aussi utilisé chez Swisscom pour leur « Bluewin TV » et, sans lui, ils ne pourraient pas fournir ce service au vu de nombre de données qu'ils doivent envoyer constamment. Là aussi, ils peuvent l'utiliser car ils possèdent tous les routeurs entre la source et les destinations. Dans le cas du SIEN, cette fonctionnalité est aussi utilisée lors de conférences vidéo et il est donc utile de regarder s'il est possible de l'implémenter en IPv6. Pour ce travail, les possibilités d'implémentations sur les équipements ont uniquement été analysées et les conclusions suivantes ont été tirées.

11.1 MLD

Les « petits » switchs L2-L3 (3750 et 3560) ne supportent pas (encore) les commandes de couches trois pour router le multicast. Par contre, ils implémentent le protocole MLD (Multicast Listener Discovery) qui est l'équivalent d'IGMP (Internet Group Management Protocol) dans IPv4, qui s'utilise en couche deux et qui est transporté par ICMPv6. Ces deux protocoles servent, dans leur version d'IP respective, à gérer, en fonction des demandes des hôtes, sur quels ports du switch doivent être envoyées les trames multicasts. La version dynamique nommée « MLD snooping » peut s'activer, soit sur l'ensemble du switch, soit pour un vlan particulier. Mais, dans tous les cas, ces switchs doivent être mis derrière un routeur qui s'occupera du routage.

11.2 Routage avec PIM

Les plus petit switchs ne supportent pas le routage du multicast comme expliqué ci-dessus. Par contre, sur le gros switch L2-L3 qui fait aussi office de routeur (le 6509), ce dernier permet, en plus du protocole MLD, de faire le routage IPv6 des paquets multicast qui ne sont pas dans une VRF. Sur cet équipement, le routage PIM (Protocol Independent Multicast) fonctionne donc mais n'est pas VRF-aware, tout comme OSPFv3 que l'on a déjà vu. Il en est de même pour le routeur 1921 pour lequel on ne peut pas avoir d'IOS avec des fonctions avancées.

Les routeurs 1841, 2811 et 7206 sont eux VRF-aware pour le multicast IPv6. Ce sont donc les seuls qui sont prêts à intégrer cette technologie sur le réseau du SIEN. On peut aussi encore noter que les deux modèles 6509 et 7206 sont aussi prêts à supporter le multicast IPv6 à travers BGP avec les mêmes commandes que pour l'IPv4. Par contre, n'ayant pas trouvé de logiciel nous servant à tester cette technologie sur notre réseau et, au vu de la limitation des équipements, nous n'avons pas essayé de l'implémenter.

12 Performances

Plusieurs cas ont été testés afin de voir comment réagissent les différents équipements du point de vue de la mémoire et du CPU. Premièrement, pour les routeurs avec une configuration IPv4 multi-VRF, le passage à IPv6 a été simulé et la mémoire a été surveillée. Ensuite, différents transferts uniquement en IPv4 puis uniquement en IPv6 et finalement, les deux en même temps, ont été effectués. Nous avons toujours essayé d'utiliser le plus de bande passante possible. Mais avec le peu de matériel à disposition, elle n'était que rarement élevée et cela peut avoir une influence sur les résultats.

Toutes les données ont été récupérées grâce au programme PRTG Network Monitor présenté au chapitre 4.4.3.

12.1 Passage au dual-stack (mémoire)

Le passage de la configuration des routeurs en dual-stack IPv4/6 multi-VRF provoque logiquement une baisse de la mémoire RAM libre. Celle-ci diffère en fonction du modèle du routeur. Différentes actions ont été analysées en détails sur le 1841 comme le montre le Tableau 3. La première remarque que l'on peut faire est que la nouvelle définition des VRF n'influence pas sur la mémoire. Cela veut dire qu'il n'y a pas de changement avec l'ancienne pour ce qui est d'IPv4, mais que la nouvelle définition sert à intégrer IPv6. L'activation du routage IPv6 avec la commande « ipv6 unicast-routing » a, par contre, un petit effet et on peut imaginer qu'il prépare de la place mémoire pour tout ce qui touche à IPv6. Par contre, quand IPv6 est activé sur une VRF, la perte mémoire n'est que minime (4kb). Cela nous montre que les VRF ne prennent pas spécialement plus de place mémoire lorsqu'on les active. Il est encore intéressant de remarquer que dès qu'on active le stack IPv6 sur une interface, cela utilise aussi un peu de mémoire mais que, par contre, les adresses supplémentaires qu'on peut lui attribuer ne changent rien, ou très peu. Il en est de même avec les routes apprises par OSPFv3 : La première coute 100kb alors que la seconde apprise n'a rien changé. On avait observé que chaque route pèse environ 124 octets, il est donc logique qu'elle ne soit pas répercutée dans nos captures. La dernière action qui a une grande influence sur la place mémoire est l'activation d'OSPFv3. Lorsque l'on en crée une nouvelle instance, le routeur a besoin d'une certaine place mémoire qui prend ici 270kb.

Action / commande dans l'ordre	Effet sur la mémoire (kb)
Transofmation des VRF (vrf upgrade-cli …)	0
ipv6 unicast-routing	-150
ipv6 cef	0
address-familiy ipv6 (VRF)	-4
ipv6 enable (sur une interface)	-50
ipv6 address (sur la même interface)	-5
ipv6 address (sur la même interface)	0
ajout d'une loopback + ipv6 enable	-10
activation d'OSPFv3 sans connexion (ipv6 router ospf…)	-270
activation d'une connexion OSPFv3 (ipv6 ospf X area Y)	-80
ajout d'une route depuis un autre routeur via OSPFv3	-100
ajout d'une route depuis un autre routeur via OSPFv3	0

Tableau 3 : Effet de différentes action sur le router 1841

Le Tableau 4 nous montre combien de kb ont été utilisés par la mémoire lors de l'activation du dual-stack sur notre petit prototype 2 qui est multi-VRF (cf. Figure 6). On voit une grande différence entre les différents modèles. Le 1841, celui qui a été étudié plus en détail ci-dessus utilise 690 kb en plus avec IPv6. Il est dans le même ordre de grandeur que le 2811 qui utilise 484kb. Le dernier routeur, le 7206 utilise plus de place (1532 kb) mais il possède toutes les routes statiques menant au 2811 ainsi que celles apprises par BGP, ce qui fait cette différence. Quant au switch L2-L3 6509 qui fonctionne ici uniquement comme un routeur, la différence est beaucoup plus impressionnante car il utilise, lui, plus de 45'000 kb. Ce nombre est plus de dix fois supérieur que celui des autres équipements. Cela peut venir du fait que ce soit un switch et qu'il va réserver de la place en mémoire pour faire le travail d'un routeur alors que les autres n'ont qu'à s'occuper du routage. Quoi qu'il en soit, tous ces équipements ont encore assez de mémoire pour apprendre plus de routes et fonctionner avec plus de VRF.

	mémoire (kb)			
Modèle	1841	2811	6509	7206
Mémoire initiale	109'190	68'602	285'594	318'102
Mémoire finale après transformation complète	108'500	68'118	239'723	316'570
Différence	-690	-484	-45'871	-1'532

Tableau 4 : Effet du dual-stack sur la mémoire

La Figure 37 nous montre le graphique sorti avec le programme PRTG Network Monitor pour le 6509. Les captures se faisant toutes les cinq minutes, on ne voit pas la descente en détail, mais on peut dire qu'IPv6 a été activé peu après 13h.

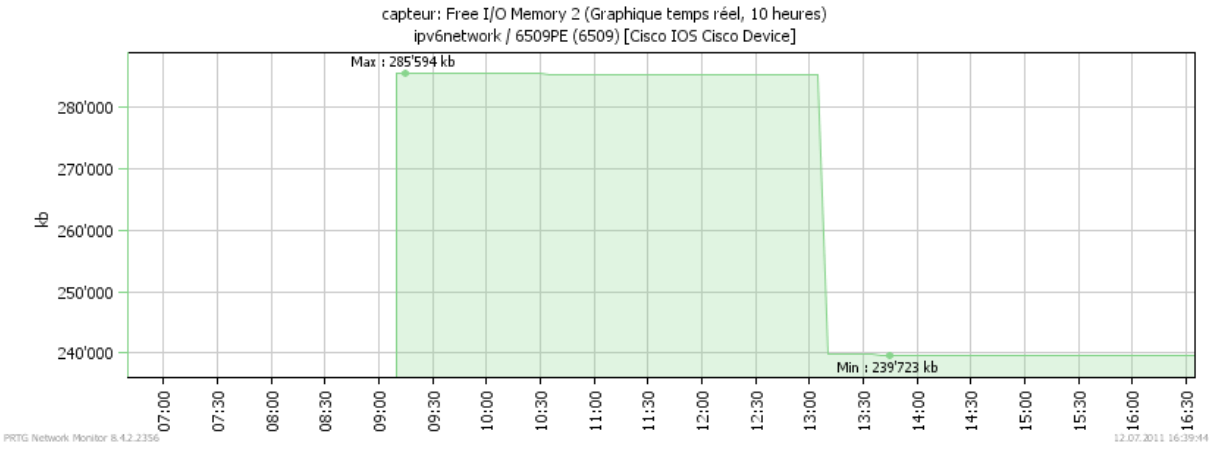

Figure 37 : Baisse de la mémoire libre du switch 6509 lors de l'activation et la mise en place d'IPv6

12.2 Transferts (CPU)

L'utilisation du CPU lors des transferts a été surprenante pour différents modèles. Pour le 1841 et le 2811, la courbe est semblable pour les deux, que ce soit en IPv4 ou en IPv6, comme nous le montre la Figure 38 et la Figure 39. Dans les deux cas, les deux interfaces sont utilisées à leur maximum étant donné qu'elles sont au 100Mb/s. L'utilisation d'IPv4 et IPv6 en même temps ne change rien aux graphiques et nous donne une utilisation similaire du CPU (Figure 40). Dans ces cas, on voit qu'il reste encore environ 50% du CPU pour les autres fonctions du routeur comme les protocoles de routage ou encore l'interface sérial. Ces graphiques devraient en réalité être plus carrés, mais la valeur récupérée par le programme est la moyenne des cinq dernières minutes ce qui lisse la courbe.

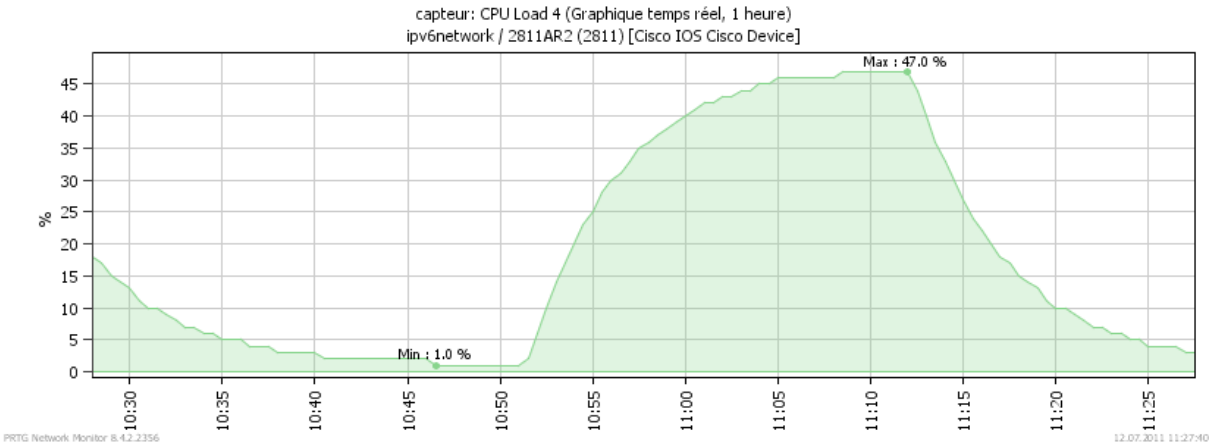

Figure 38 : Utilisation du CPU lors de transferts en IPv4 uniquement sur un 2811

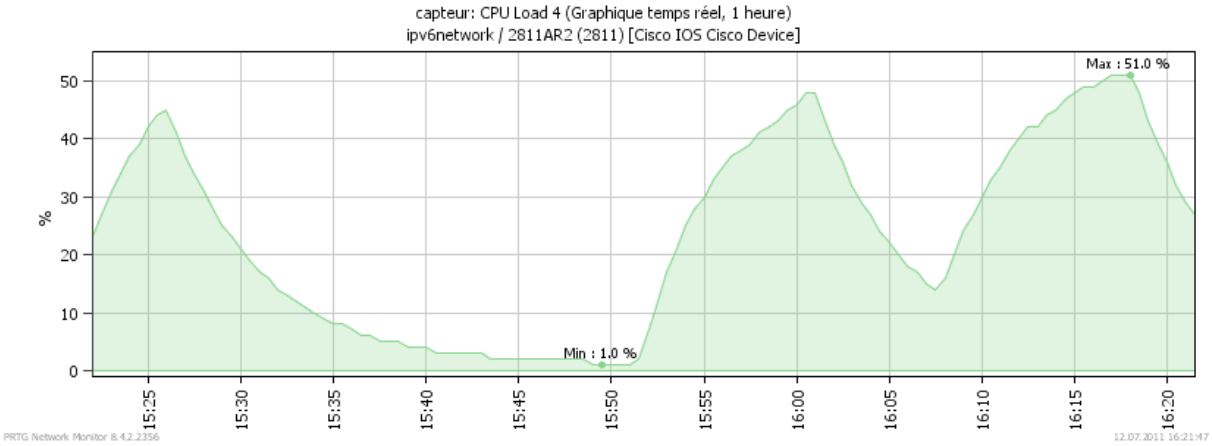

Figure 39 : Utilisation du CPU lors de transferts en IPv6 uniquement sur un 2811

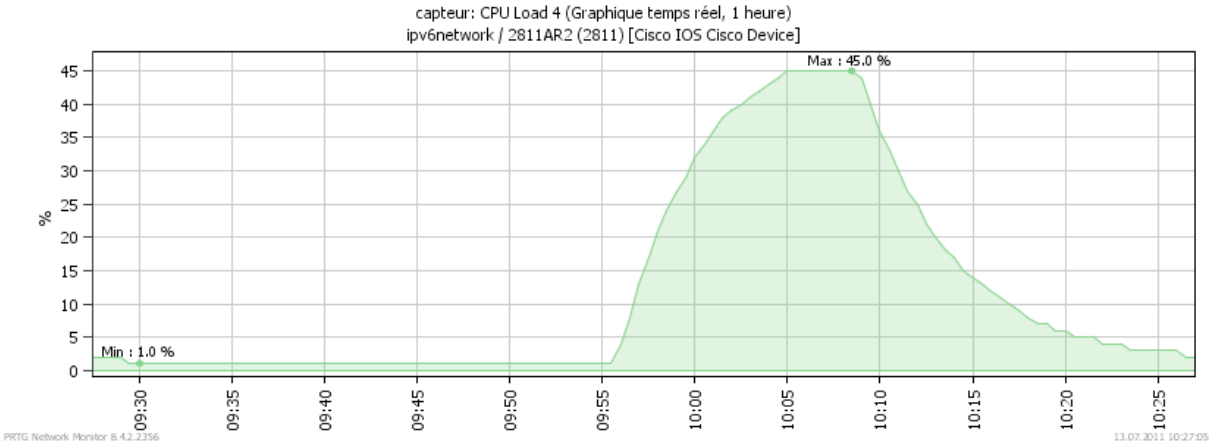

Figure 40 : Utilisation du CPU lors de transferts en IPv4 et IPv6 simultanément sur un 2811

Avec les deux autres routeurs (1921 et 7206), on peut tirer les mêmes conclusions mais avec un autre pourcentage ce qui est logique vu qu'ils doivent être capables de gérer bien plus de trafic. Pour le 1921 on se trouve environ à 13% et le 7206 à 8% (Figure 41) avec toujours 100Mb/s de transfert.

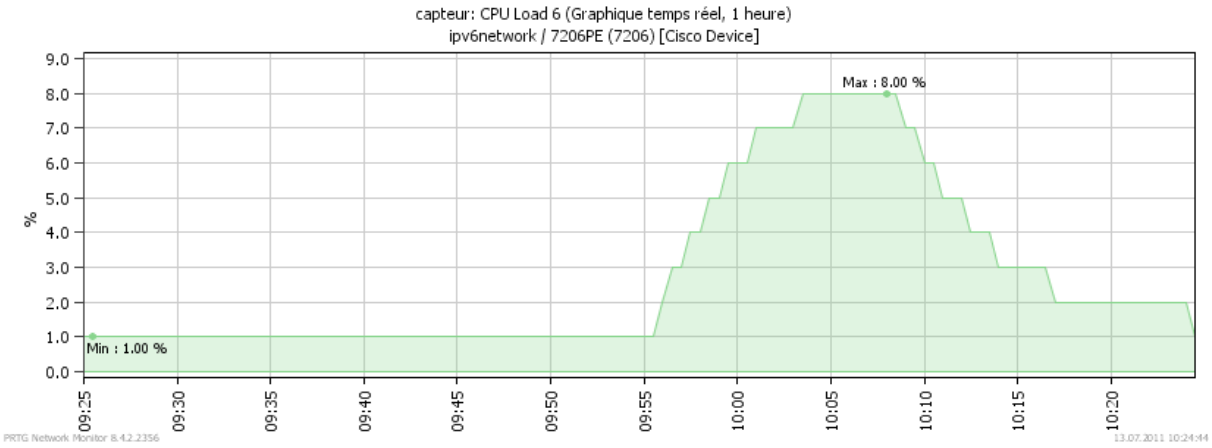

Figure 41 : Utilisation du CPU lors de transferts en IPv4 et IPv6 simultanément sur un 7206

Pour les switchs L2-L3, on trouve plusieurs cas différents mais tous sont surprenants. Sur le 6509, rien ne bouge. Cela veut dire que les deux CPU monitorés restent à 1% d'utilisation quel que soit le transfert. Sur le switch 3750G, c'est pareil mais on reste à 5% d'utilisation continuellement. Le CPU du 3560 est, lui, à 23% d'utilisation au repos et lorsqu'un transfert est lancé, il baisse jusqu'à 8% dans certains cas comme le montre la Figure 43 lors d'un transfert illustré sur la Figure 42 (en bleu l'envoi, en orange la réception et en vert la somme des deux). Afin de s'assurer que cette observation soit correcte, nous avons regardé ces valeurs directement sur le switch et il indique les mêmes chiffres. Là aussi, les comportements sont les mêmes que ce soit en IPv4, en IPv6 ou avec des transferts mixtes.

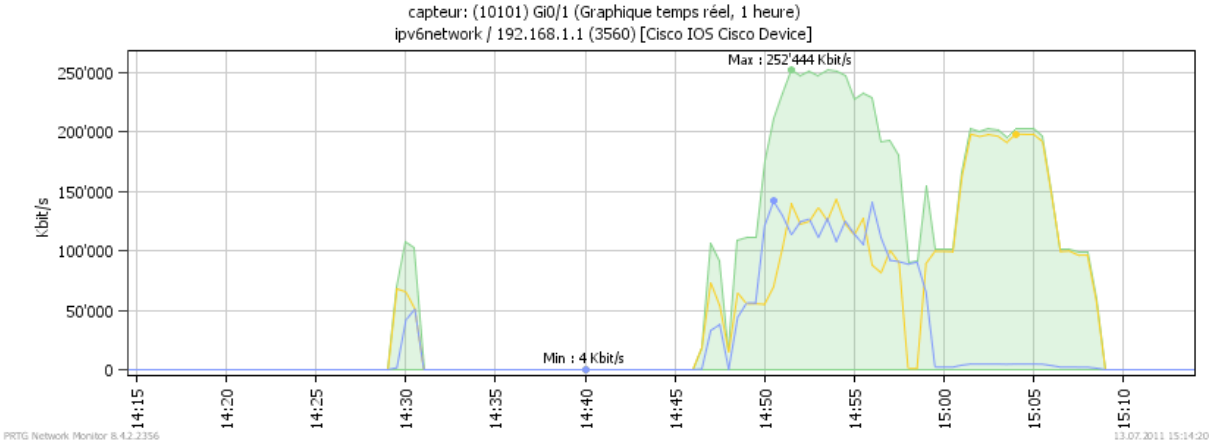

Figure 42 : Débit des transferts effectués à travers le switch 3560 en IPv6 uniquement

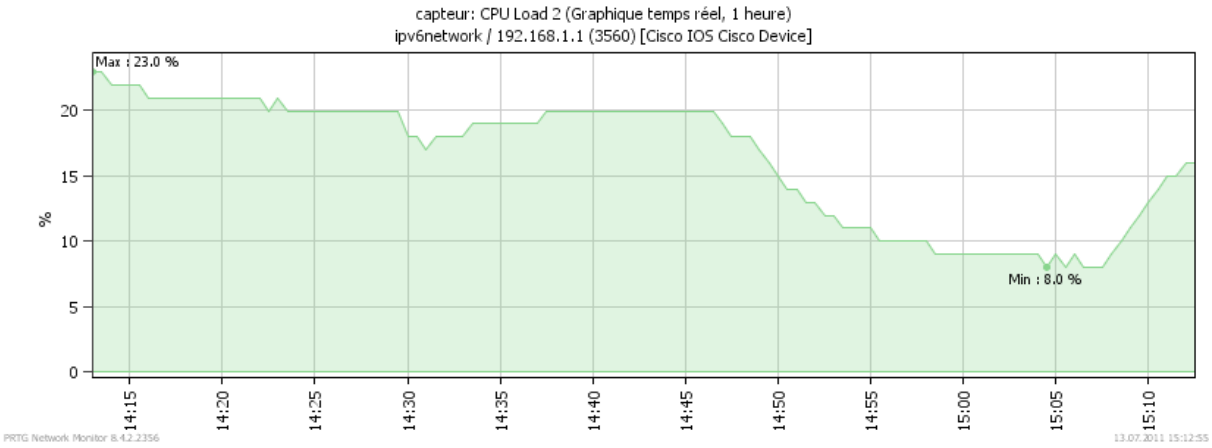

Figure 43 : Utilisation du CPU lors de transferts en IPv6 uniquement sur un 3560

On a donc pu remarquer qu'aucun modèle n'était dépassé par IPv6. Pour les routeurs cela semble correcte, mais pour les switchs, il est fort possible que cela vienne du fait que l'on n'ait uniquement utilisé une petite partie de leur capacité. Quoi qu'il en soit, pour l'utilisation qui en a été faite, IPv6 semble aussi bien intégré en termes de performances que l'est IPv4. Il faudrait tout-de-même procéder à quelques tests plus approfondis et attendre les chiffres officiels de Cisco avant de mettre un de ces équipements dans un réseau de production IPv6.

13 Conclusion

Ce travail a permis de remarquer au fur et à mesure des tests et des lectures qu'IPv6 a vraiment été fait dans le but d'améliorer IPv4. On y retrouve toutes les fonctionnalités de ce dernier avec plusieurs améliorations intéressantes. L'exemple le plus flagrant est l'attribution des adresses car, en plus du mode DHCP comme on le connait en IPv4, il en existe deux nouveaux (stateless et statefull), qui devraient lui prendre la place, car ils simplifient énormément le serveur DHCP. Le protocole IPv6 est un concept bien réfléchi basé sur l'expérience de nombreuses années de pratiques en IPv4, les difficultés vont principalement venir des implémentations (Cisco ou autres), qui sont actuellement encore lacunaires.

Si l'on regarde maintenant le sujet du travail qui est l'implémentation d'un réseau dual-stack sur des équipements Cisco, on remarque que la base d'IPv6 y est bien implémentée mais que certaines parties plus complexes ou moins utilisées ne sont pas encore disponibles sur les IOS testés. Il n'y a donc aucun problème à faire un réseau simple, utiliser les nouvelles fonctions que l'on ne connaissait pas en IPv4, comme la configuration automatique des adresses ou les différents modes DHCP. Le protocole ICMPv6 est lui-aussi totalement fonctionnel ce qui est presque une obligation étant donné qu'il est la base d'IPv6. Par contre, dès que l'on s'attaque au routage utilisant des fonctions avancées incluent dans IPv4, comme l'OSPF multi-VRF ou encore un backbone MPLS-VPN, ces dernières deviennent très limitées en IPv6. Comme on l'a vu, on ne peut pas avoir un backbone MPLS tournant totalement sur IPv6, l'IGP de ce dernier doit être en IPv4. Ou encore, à l'heure où ces lignes sont écrites, OSPFv3 ne gère pas le multi-VRF, ce qui bloque toute la transition du SIEN vers IPv6, car leur réseau utilise une multitude de VRF. Donc, même si Cisco laisse penser que son matériel est totalement prêt à accueillir IPv6, il faut tout-de-même tester en profondeur toutes les fonctionnalités que l'on veut utiliser afin de s'assurer qu'elles soient totalement opérationnelles.

Dans le cas précis du SIEN, je leur recommande de préparer à l'avance tout ce qui est déjà possible de faire sans toucher au matériel. Cela veut dire l'adressage, le choix des modes DHCP pour les différents sous-réseaux ou encore la gestion de la sécurité. Il est important de le faire quand on a une vue d'ensemble, afin d'éviter de rater ce que l'on ne verrait pas en s'occupant uniquement d'un sous-réseau. Une fois cette partie théorique effectuée, on a une bonne base pour planifier la transition.

Ensuite, je pense qu'il est important de se familiariser avec IPv6 en créant un petit réseau de test ressemblant au prototype 1, afin de pouvoir observer la réaction des différentes commandes, car c'est en essayant que l'on apprend et comprend le mieux.

Pour le réseau de production en lui-même, je propose d'utiliser l'approche « core-to-edge » et donc de commencer par s'occuper des routeurs du backbone. L'avantage de cela est que l'on n'a pas besoin d'OSPFv3 multi-VRF et que l'on peut utiliser la technique d'intégration 6VPE pour y arriver. Cela doit se faire sans aucune répercussion pour le réseau existant car il n'y a encore aucune route à échanger, mais tout sera prêt pour le faire.

Une fois que ces derniers sont prêts pour IPv6, du temps aura passé et on peut espérer que Cisco ait implémenté OSPFv3 sur les interfaces VRF. A ce moment-là on pourra intégrer tous les CE qui sont prêts à le supporter (on sait qu'il faut faire principalement attention à la mémoire) au réseau IPv6. Dans le cas contraire, il faut utiliser le routage statique afin d'intégrer tout de même quelques CE

pour faire des tests sur le réseau de production. L'étape d'après se fera par zone, principalement en fonction des compatibilités matérielles. Dans celles où ce sera possible, on va implémenter IPv6. Le rythme de transition des extrémités du réseau dépendra donc de plusieurs facteurs dont les capacités des équipements ou le besoin d'IPv6, car il y a fort à parier que certaines applications vont demander IPv6 avant que l'on ne soit prêt.

De mon point de vue, IPv6 n'est pas assez connu aujourd'hui car beaucoup de personnes en ont peur et pense qu'il n'est pas utile. Mais, à force de travailler avec, on remarque rapidement toutes les possibilités supplémentaires qu'il apporte. Si je devais, aujourd'hui, créer un nouveau réseau, je le ferai directement en IPv6 si c'est possible. Le fait de devoir mettre des adresses globales uniquement sur les interfaces des routeurs, qui sont dans un domaine de collision qui comprend des hôtes, facilite énormément les configurations. Mais ce n'est pas le seul avantage, même si ce ne sont des détails, ils deviennent très agréables à force d'utilisation.

Ce travail m'a permis de bien me familiariser avec tout le matériel et l'univers Cisco. Le fait, d'avoir plus de liberté que lors de laboratoires, oblige une plus grande implication au niveau de la recherche d'informations, ce qui permet de cumuler beaucoup de connaissances que l'on n'utilisera peut-être même pas dans ce projet, mais qui seront surement utiles une fois dans le futur. Le fait d'arriver à une solution fonctionnelle à la fin est toujours un plus.

Je peux m'imaginer et j'aimerais que ce travail puisse m'aider dans ma carrière professionnelle future grâce aux connaissances acquises. Que ce soit des connaissances générales sur IPv6 ou la manière de l'aborder sur des équipements réseaux, cela pourra être fort utile vu que peu de personnes ont de l'expérience sur ce sujet. Sur le plan personnel, étant donné qu'il se trouve dans un domaine qui me plait énormément, il a été effectué avec beaucoup de motivation et de plaisir. Je pense donc avoir atteint les objectifs demandés et j'espère que mon travail va pouvoir aider le SIEN comme il l'espérait dans la découverte de ce nouveau protocole et la migration vers celui-ci.

14 Remerciements

Je tiens à remercier particulièrement M. Jérôme Vernez qui m'a suivi pendant toute la période du travail et qui était toujours là pour m'aider à résoudre des problèmes. J'aimerai aussi remercier M. Stephan Robert de m'avoir fait confiance en m'attribuant ce travail et pour ses conseils pour la rédaction du rapport. Pour la partie technique, M. Fabien Bruchez a toujours pris le temps de répondre à mes questions et c'est pourquoi j'aimerai aussi le remercier.

Ma maman, Elisabeth, et ma sœur, Joanne, m'ont été d'une grande aide pour la relecture même si le sujet leur était totalement inconnu. Merci donc à elles pour le temps pris à me relire. Finalement, un dernier merci à ma copine, Joëlle, qui a premièrement m'a soutenue pendant les parties critiques de mon travail et qui, deuxièmement, m'a donné des conseils pour la mise-en-page.

15 Liste des références

[1] Citation de Mme Silvia Hagen lors de sa présentation (28 juin 2011)
[2] Citation de Mme Silvia Hagen lors de sa présentation (28 juin 2011)
[3] http://fr.wikipedia.org/wiki/Canton_de_Neuch%C3%A2tel (juillet 2011)
[4] http://www.slideshare.net/feb_989/ipv6-mpls-by-patrick-grossetete, slide 5 (juillet 2001)
[5] Explications de Mme Silvia Hagen lors de sa présentation (28 juin 2011)

16 Sources

16.1 Bibliographie

- Cours de PDR 2010, La Couche Réseau, Stephan Robert, HEIG-VD
- Cours IPv6 2010, Fabien Bruchez, LANExpert SA
- IPv6 Essentials, Silvia Hagen, 2006
- Modes d'emploi et listes des commandes, Cisco

16.2 Webographie

16.2.1 IPv6 général

- http://www.cisco.com/ipv6/
- http://www.wikipedia.org/
- http://www.6diss.org/e-learning/index.html
- http://www.scribd.com/doc/7399715/IPV6-for-CCNA
- http://computernetworkingnotes.com/ccna_certifications/ipv6_neighbor_discovery.htm

16.2.2 OSPF

- https://www.racf.bnl.gov/Facility/TechnologyMeeting/Archive/06-30-04-CISCO/Using-OSPF-in-MPLS-VPN-Environment.pdf
- http://packetlife.net/blog/2008/jun/24/ospf-area-types/
- http://www.ciscomadesimple.be/2010/11/15/gns3-ospf-lab-2eme-partie/
- http://www.scribd.com/doc/55938127/49/Multiple-OSPFv3-Instance-Support

16.2.3 Backbone

- http://ieoc.com/forums/t/11808.aspx (VRF)
- http://www.tebyan.net/index.aspx?pid=31159&BookID=22012&PageIndex=51&Language=3
- http://www.ipv6-taskforce.gr/events/users2011/03-IPv6forEnterprise.pdf
- https://learningnetwork.cisco.com/thread/8922?start=15&tstart=0
- http://www.slideshare.net/feb_989/ipv6-mpls-by-patrick-grossetete
- http://www.ine.com/all-access-pass/training/playlist/ccie-routing-switching-advanced-technologies-class/mpls-layer-3-vpn-11000.html

17 Glossaire

ABR	Area Border Router
ADSL	Asymmetric Digital Subscriber Line
ARP	Address Resolution Protocol
BDR	Backup Designated Router (OSPF)
BGP	Border Gateway Protocol
CE	Customer Edge (router)
CEF	Cisco Express Forwarding
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DR	Designated Router (OSPF)
DUID	Demand Unique Identifier
EIGRP	Enhanced Interior Gateway Routing Protocol
HEIG-VD	Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud
ICMP	Internet Control Message Protocol
IETF	Engineering Task Force
IGMP	Internet Group Management Protocol
IOS	Internetwork Operating System
IP	Internet Protocol
LDP	Label Distribution Protocol
MAC	Media Access Control
MLD	Multicast Listener Discovery
MPLS	MultiProtocol Label Switching
MTU	Maximum Transmission Unit
ND(P)	Neighbor Discovery Protocol
OSPF	Open Shortest Path First
P	Provider (router)
PDR	Protocoles de Réseaux
PE	Provider Edge (router)
PIM	Protocol Independent Multicast
PRTG	Paessler Router Traffic Grapher
RAM	Random Access Memory
RIR	Regional Internet Registry
SDM	Switch Database Management
SIEN	Service Informatique de l'Entité Neuchâteloise
SIG	Services Industriels de Genève
TDP	Tag Distribution Protocol
TFTP	Trivial File Transfer Protocol
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding

18 Tables

18.1 Tableaux

18.2 Figures

Annexes

Images

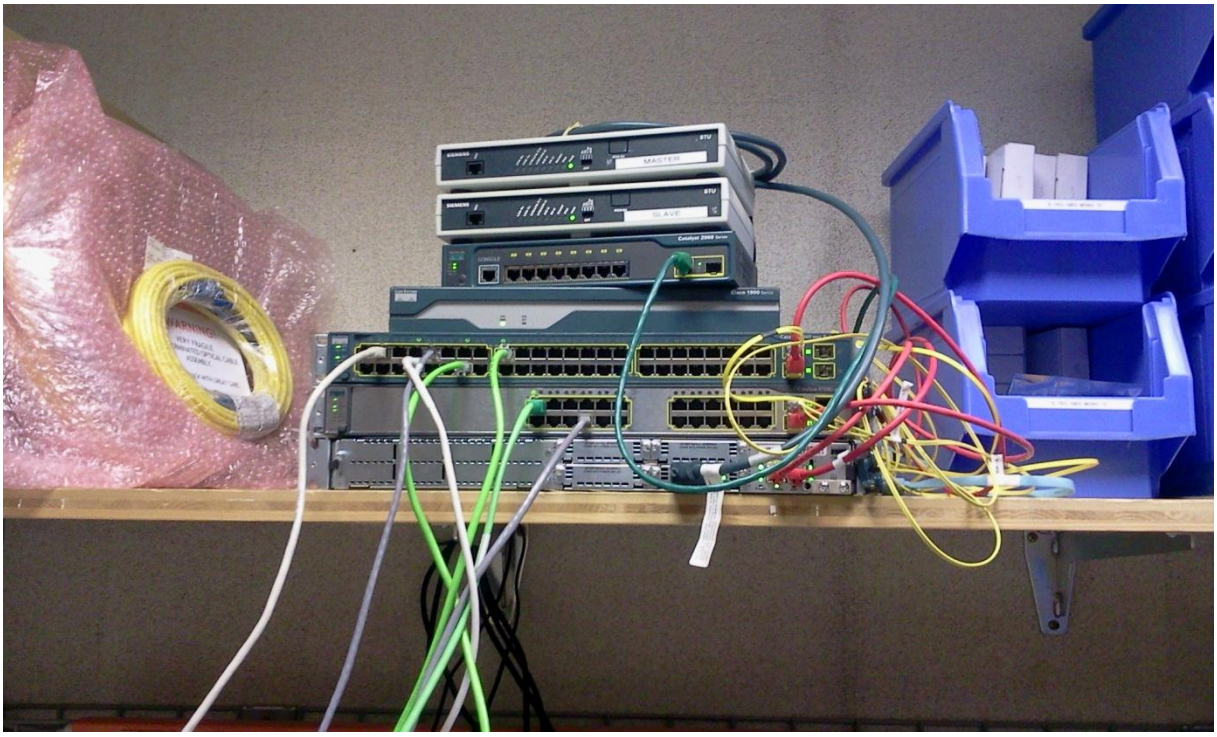

Image a : Montage du prototype 1

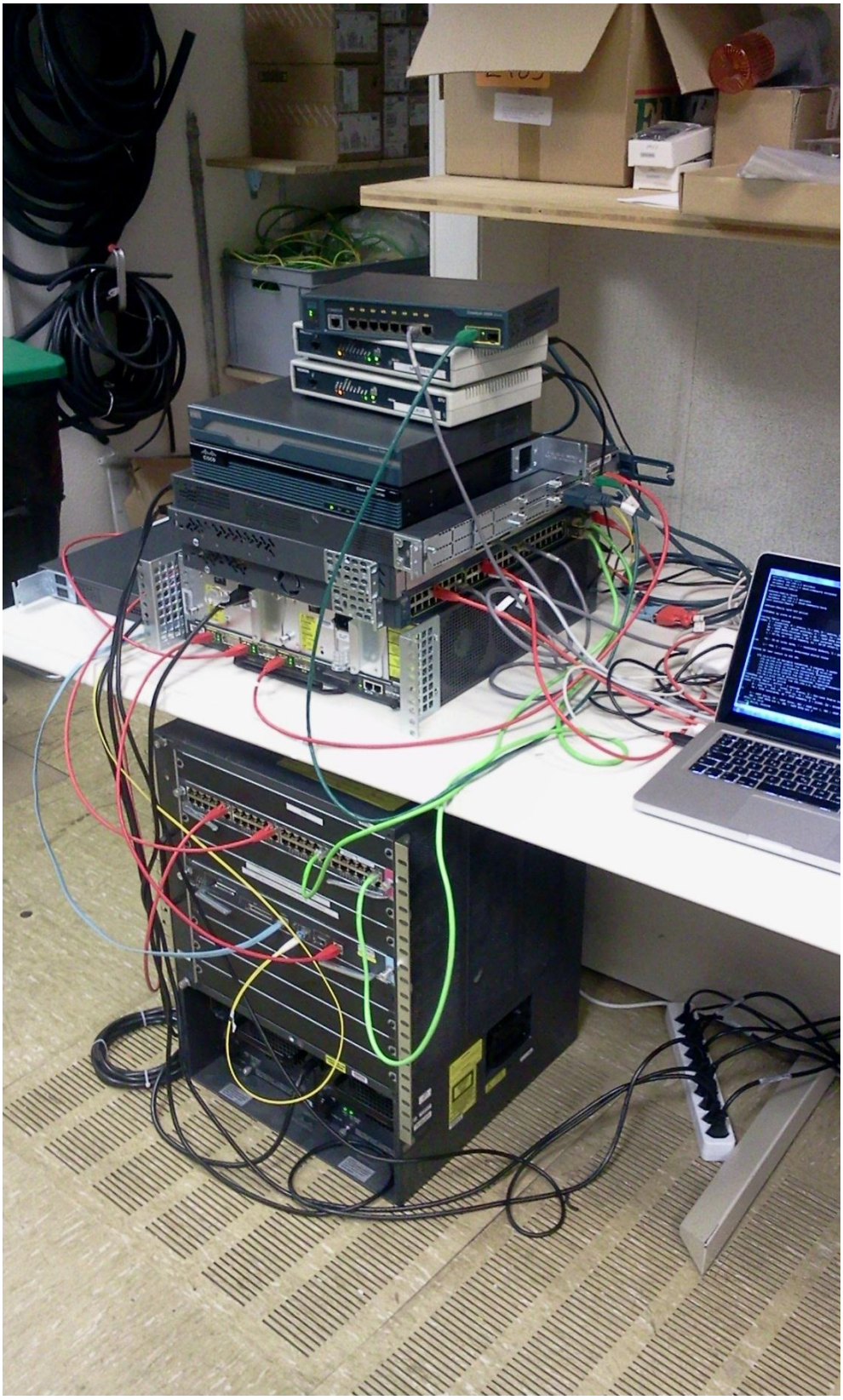

Image b : Montage du prototype 2

Journal de travail

Le journal de travail a été construit de la manière suivante : Etant donné qu'un jour par semaine était réservé exclusivement à ce travail et que nous nous retrouvions au SIEN, le travail fait ces jours précis a été noté et on y a toujours passé huit heures sauf indication contraire. En dehors de ça, du travail a aussi été fait à la maison. Celui-ci est indiqué dans « Entre deux » et indique aussi le nombre d'heures passées à le faire.

Avant : Recherche de documentation
Lecture du cours de PDR sur IPv6

16 janvier : Premier rendez-vous au SIEN, Neuchâtel : Discussion du projet, présentation du SIEN.

Entre deux : Lectures & vidéos sur IPv6 (6h)

23 février : Au SIEN :
Etablissement du prototype1
Plan d'adressage
Découverte du matériel (routeurs et switchs)
Mise-à-jour de l'IOS des switchs

Entre deux : Lectures sur IPv6
Recherche des commandes vlan et ospf (IPv4) dans les manuels (6h)

9 mars : Au SIEN :
Mise-à-jour du logiciel des routeurs
Monté les 2/3 du réseau

Entre deux : Recherche sur IP Helper pour l'optimiser (4h).

16 mars : Au SIEN :
Fin du montage du réseau et observé comment réagit OSPF (1/2 journée)

Entre deux :

23 mars : Au SIEN :
Réglé le problème de l'attente pour la récupération de l'adresse IP (STP)
Début de la partie IPv6 :
Fait l'architecture du réseau IPv6
Configuré IPv6 pour les deux premier sous-réseaux (DHCP stateless et OSPF)
- Ping et telnet fonctionnent avec IPv6.

Entre deux : Recherche documentation sur les différentes options du dual-stack. (2h)

30 mars : Au SIEN :
Intégré la dernière partie du réseau à IPv6.
- OSPF, ping et telnet fonctionnent en IPv4 et IPv6 (Dual stack)

Lecture des différents mode-d'emplois pour trouver des informations
Recherches pour l'utilisation du DHCPv6

- Recherché et utilisé la capture intégrée aux routeurs (pas switchs)
- Recherche des différentes possibilités d'utiliser DHCPv6

Entre deux : Discussion des problèmes de logique IPv6 avec M. Bruchez. Réflexions sur que changer et comment (2h)

06 avril : Au SIEN :
Amélioration du réseau IPv4 et IPv6 sur les points suivants :
- Ajout de loopback utiles pour la gestion (adresses routées)
- Ajout des réseau P-à-P en IPv6 pour pouvoir pinger les interfaces
- Optimisation d'OSPF

Résolution des problèmes DHCPv6 statefull. Tout fonctionne à présent. On va pouvoir se concentrer sur les subtilités IPv6 à présent.

Entre deux : Lecture du cours IPv6 (4h)

13 avril : Au SIEN :
Présentation de ce que l'on a fait à Jérôme
Lecture du cours IPv6
Fait schéma IPv6
Configuration DNS sur les switchs
Rencontre avec Stephan Robert

Entre deux : Lecture du cours IPv6 + test sur le matériel de l'école (6h)

20 avril : Au SIEN :
Création des différents fichiers de configuration pour passer rapidement du statefull au stateless au besoin.
Différents tests en statefull et stateless
Recherche de mise-à-jours des équipements Cisco
Recherche comment analyser la mémoire et sortie des premiers chiffres

Entre deux : Transfert des résultats dans un fichier excel, premières analyses (4h)
Fin lecture du cours IPv6 (3h)

4 mai : Au SIEN :
Analyse des mémoires des switchs et routeurs :
Taille (octet) d'une route IPv6 sur nos switchs et sur nos routeurs
Place réservée aux routes IPv6 sur les switchs L3
Comparaison avec le nombre de routes IPv4 sur les switchs en production
Recherches pour faire que les équipements s'inscrivent dans le DNS pour les joindre plus facilement

Entre deux :

11 mai : au SIEN :
Mise à jour du journal de travail
Suite recherche pour inscription des équipements au DNS : il faut le mettre

manuellement, mais c'est d'un grand service.
Commencement du rapport

Entre deux : Ecriture du rapport (3h)

18 mai : Au SIEN :
Ecriture du rapport
Vérification de l'utilisation du routeur lors d'envoi de paquets intra-sous-réseau
Réunion avec Stephan Robert

Entre deux : Rédaction du rapport (4h)

25 mai : Au SIEN :
Modification de la configuration d'un routeur pour y introduire les VLAN.
Rédaction du rapport
Mise-à-jour des schémas de réseau

Entre deux : Rédaction du rapport (3h)

1er juin : A la HEIG-VD :
Rédaction du rapport

Entre deux : Rédaction du rapport (5.5h)

8 juin : Au SIEN :
Quelques tests de mémoire sur les routeurs avec OSPF
Prise de captures manquantes pour le rapport
Rédaction du rapport

Entre deux : Rédaction et mise en page du rapport (9h)

15 juin : Au SIEN :
Rédaction du rapport intermédiaire

Entre deux : Mise en page du rapport (9h)

17 juin : Rendu du rapport intermédiaire

20 juin : Au SIEN :
Découverte du matériel
Schéma du Prototype 2
Début upgrade du matériel

21 juin : Au SIEN :
Upgrade du matériel
Compréhension des VRF
Début de la configuration

22 juin : Au SIEN :
Configuration OSPF multi-area et multi VRF
Rencontre avec Stephan Robert

23 juin : Au SIEN :
Finalisation de la configuration OSPF
Début configuration MPLS/BGP

24 juin : Au SIEN :
Ajout du RR pour BGP
Débug BGP (il faut indiquer quelles interfaces doivent tagguer les paquets)
Création d'un réseau de gestion

26 juin : A la maison (4h):
Fait divers modifications demandée après le rapport intermédiaire
Fait le schéma du réseau sur Visio
Lecture des fichiers donnés par M. Amann

27 juin : Au SIEN :
Terminé tout IPv4 avec le réseau de gestion
Début IPv6 :
Plan d'adressage
Tentative de faire un réseau OSPF multi-vrf, mais OSPFv3 n'accepte pas cette option.

28 juin : A la maison :
Recherches et lectures sur OSPFv3, VRF, Backbone MPLS
Au SIG (Genève) :
Séminaire sur IPv6 avec Mme Hagen

29 juin : Au SIEN :
Recherches sur OSPFv3 VRF-aware
Recherches sur EIGRP VRF-aware
Tests de EIGRP VRF-aware
Abandon du Backbone multi-VRF pour un mono-VRF car pas possible

30 juin : Au SIEN :
Installation du Backbone IPv6 mono-VRF

1er juillet : Au SIEN :
Tests sur le Backbone qui ne tag pas les paquets
Recherche d'une solution pour tout de même tester un Backbone.
Installation du même backbone (simlifié) en IPv4 mono-VRF

3 juillet : A la maison :
Rédaction du rapport

4 juillet : Au SIEN :
Recherche des nouvelles versions d'IOS
Nouveau test d'implémentation du backbone IPv6

5 juillet : Au SIEN :
Installation du backbone 6PE
Captures sur ce backbone

6 juillet : Au SIEN :
Créé la configuration parfaite 6PE
Fait des captures des tables de routage IPv6
Rédaction du rapport

7 juillet : A la HEIG-VD :
Fait les changements demandés à la table des matières
Fait le glossaire
Mis en page du texte écrit

8 juillet : A la HEIG-VD le matin et à la maison le reste de la journée :
Rédaction du rapport

10 juillet : A la maison :
Analyse des tables de routages VRF/BGP/MPLS
Rédaction du rapport

11 juillet : Au SIEN :
Rencontre avec Stephan Robert
Nouveau tests de EIGRP
Test des VRF IPv6 sur les switchs
Installation de PTRG pour SNMP

12 juillet : Au SIEN :
Différents tests avec SNMP :
IPv4 seulement
Ajout d'IPv6
IPv6 seulement

13 juillet : Au SIEN :
Suite des tests SNMP :
IPv4 & IPv6
Comparaisons avec des routeurs de dernière génération
Performances en IPv6 sur les switchs L2-L3
Recherche des différences avec les nouveau matériels

14 juillet : A la HEIG-VD :
Première relecture de ce qui a été écrit
Rédaction du rapport
Mise-à-jour des plans d'adressage

15 juillet : Au SIEN :
Derniers tests de performance
Test des commandes pour le multicast
Captures utiles au rapport

17 juillet : A la maison :
Rédaction du rapport
Relecture des titres des figures

18 juillet : A la maison :
Rédaction du rapport
Mise en page des tableaux de commandes

19 juillet : A la maison :
Rédaction des dernières partie du rapport
Impression

20 juillet : A la maison :
Relecture personnelle du rapport

21 juillet : A la maison :
Corrections
Relecture par quelqu'un d'autre
Rédaction de la conclusion

22 juillet : A la maison :
Rédaction de la conclusion

24 juillet : A la maison :
Correction des erreurs trouvée dans le rapport par les relecteurs
Mise-en-page en fonction des directives reçues

25 juillet : Au SIEN :
Rédaction du résumé
A la maison :
Suite de la correction des erreurs trouvées par les relecteurs

26 juillet : A la maison :
Inclusion des sources
Gestion des références
Mise-en-page finale

27 juillet : Impression
Gravure du DVD
Rendu du travail