

Utilisation d'OpenFlow et des modules Split Data Plane de DELL pour traiter le DUID-MAC-spoofing des requêtes DHCPv6

Marc Bruyère

Laboratoire d'Analyse et d'Architecture des Systèmes / CNRS UPR8001
Complexe Scientifique de Rangueil
7, avenue du Colonel Roche
31400 Toulouse

David Delavennat

Centre de Mathématiques Laurent Schwartz / CNRS UMR7640
Bâtiment 6, Ecole Polytechnique
91128 Palaiseau

Résumé

IPv6 a longtemps été associé à un mécanisme d'auto-configuration sans-état (Router Advertisements). La mise à disposition tardive d'une implémentation de protocole d'attribution d'adresses IPv6, avec-état, par l'ISC, a été un frein à son déploiement sur un parc de postes clients « administrés ». Cependant, l'arrivée du protocole DHCPv6 ne se fait pas sans soulever de nouvelles difficultés opérationnelles. Dans les environnements à double pile IP, le fait que DHCPv4 et DHCPv6 n'utilisent pas les mêmes attributs pour identifier les objets IP (MAC dans le premier cas, DUID dans le second), complique la tâche des ASR (plusieurs variantes de DUID coexistent) et rend problématique la corrélation des adresses au sein des journaux d'activités réseaux.

Une implantation de serveur DHCPv6 est disponible, autorisant le référencement d'objets IP par leurs adresses MAC, i.e. ne tenant pas compte de l'attribut DUID. Nous souhaitons toutefois pouvoir utiliser notre infrastructure DHCPv6 ISC préexistante, qui, elle, ne supporte que les attributs DUID, tout en référençant nos postes clients par leurs adresses MAC, peu importe la version du protocole DHCP.

Cette contrainte peut être résolue en utilisant un mécanisme embarqué au sein des modules Split Data Plane des commutateurs DELL SDN de nouvelle génération. Ces modules embarquent des applicatifs métiers au sein des commutateurs afin de réaliser des opérations au plus près du réseau.

Mots-clés

IPv6, DHCPv6, DUID, MAC, Network-Processor, NetFPGA, Software Defined Network

1 Introduction

Début 2012, la question s'est posée de déployer IPv6 sur le parc informatique du Centre de Mathématiques Laurent Schwartz de l'Ecole Polytechnique. DHCPv6 s'est imposé face à l'auto-configuration IPv6 afin de garder la maîtrise des adresses IP distribuées.

Très vite, il est apparu que la conception du protocole était incompatible avec l'usage opérationnel satisfaisant, qu'il était souhaitable de transposer de DHCPv4 à DHCPv6.

Afin de trouver une solution, une maquette de sniffer L2-L3 a été réalisée capable de traiter le spoofing du DUID-client des requêtes DHCPv6. Une fois le modèle validé, des contacts ont été pris avec plusieurs constructeurs de matériel réseau dans le but d'implémenter cette fonctionnalité dans les matériels d'extrémité. Ils sont en effet les plus à même de voir passer l'adresse MAC des interfaces réseaux émettrices des requêtes. DELL a répondu favorablement en proposant de traiter ce besoin sur leurs commutateurs SDN de nouvelle génération.

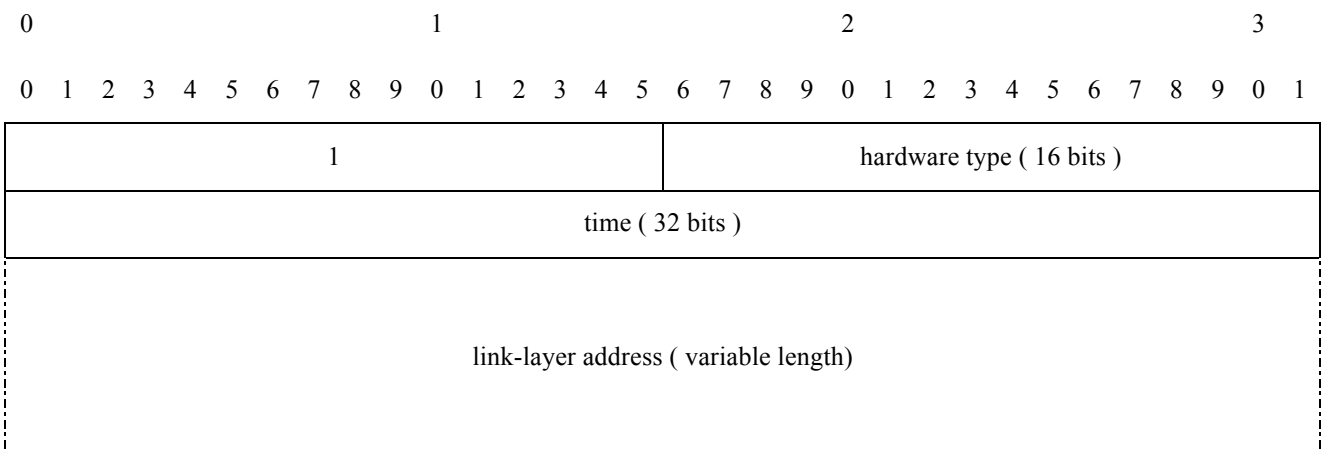
2 Présentation générale

2.1 DHCPv6

Le protocole DHCPv6 définit un attribut DUID (DHCP Unique Identifier) pour identifier clients et serveurs. Cet attribut doit être vu de manière opaque ; sa valeur seule intervenant lors de tests de comparaison.

La RFC3315 (Juillet 2003) définit trois sortes de DUID (DUID-LLT, DUID-EN et DUID-LL), tout en spécifiant que de nouveaux types peuvent voir le jour dans le futur. C'est le cas avec la RFC6355 (Aout 2011) qui introduit le DUID-UUID.

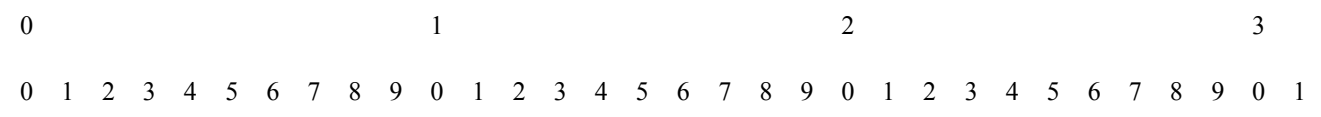
2.1.1 DUID-LLT

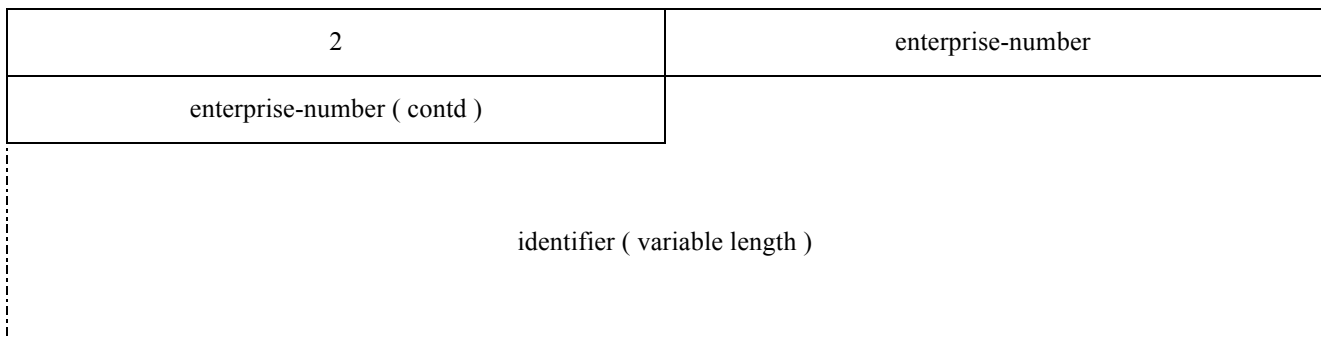


Contrairement à ce que l'on pourrait penser, par analogie avec le protocole DHCPv4, le champ « link-layer address » ne contient pas l'adresse MAC de l'interface réseau émettrice de la requête. C'est celle de l'une, arbitraire, des interfaces réseau, connectée au client DHCP à l'installation du poste client (lors de la génération du DUID). Le même DUID-LLT est utilisé pour configurer l'ensemble des interfaces réseau.

La stabilité du DUID-LLT (si l'on fait abstraction de la réinstallation des machines) impose un stockage local au client DHCPv6 et n'est donc pas adapté au boot en réseau.

2.1.2 DUID-EN

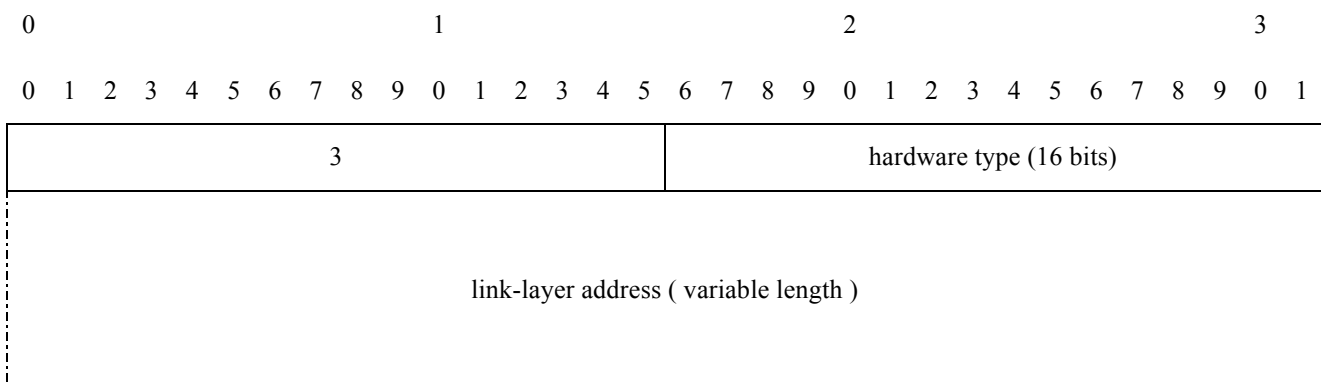




Le DUID-EN est constitué d'un préfixe PEN (Private Enterprise Number) constructeur enregistré auprès de l'IANA suivi d'un identifiant unique assigné par ce même constructeur selon une règle qui lui est propre.

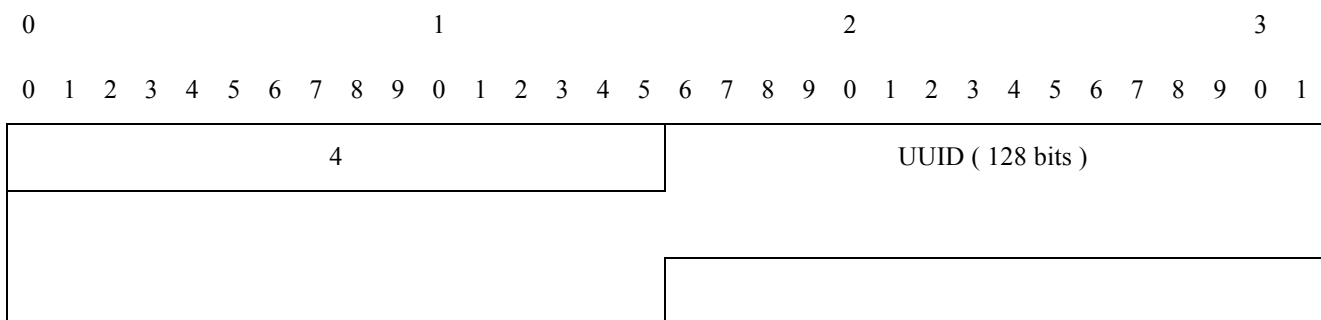
Le DUID-EN devrait être enregistré sur un support local au client et non effaçable.

2.1.3 DUID-LL



Le DUID-LL se distingue du DUID-LLT par le fait que le champ « link-layer address » contient l'adresse MAC de l'une, arbitraire, des interfaces réseaux connectée de manière permanente au client DHCP.

2.1.4 DUID-UUID



Avec le temps, est apparu le fait que les trois DUID précédant n'étaient pas stables vis-à-vis d'un matériel donné.

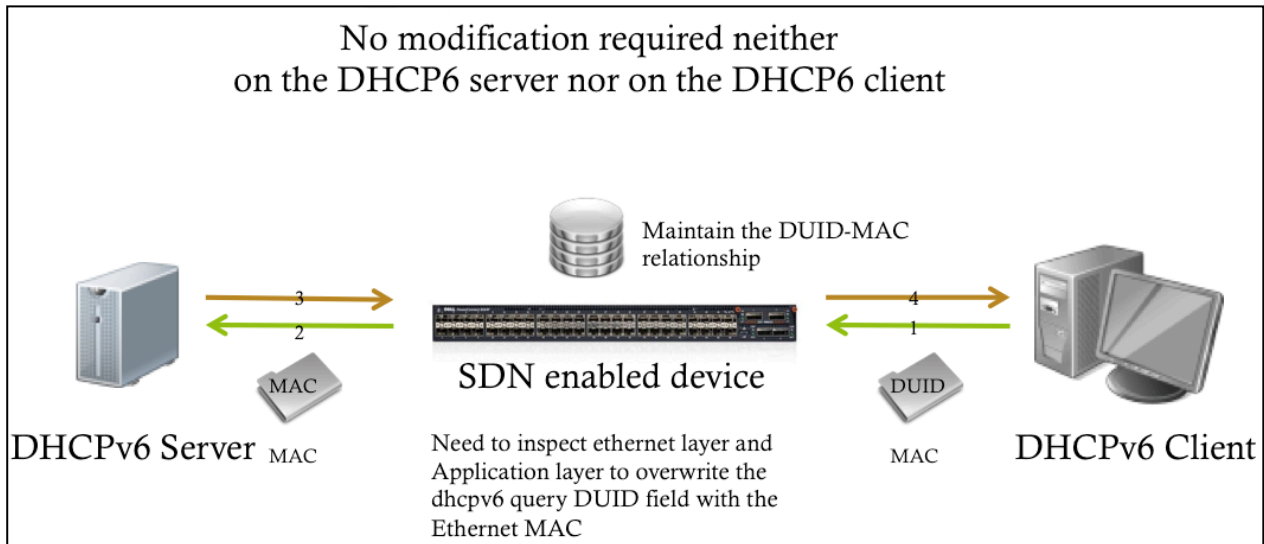
Le DUID-UUID a donc été défini en précisant que l'UUID choisi doit être :

1. accessible aux firmwares de démarrage réseau
2. persistant tout au long des phases de redémarrage et de configuration d'un boot réseau.

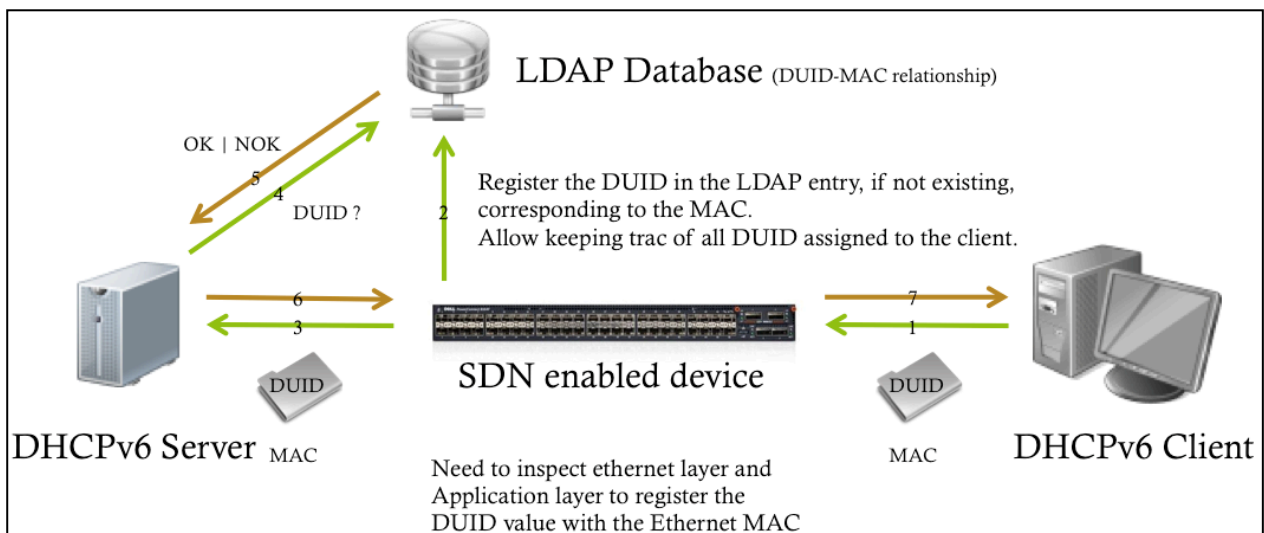
Dans tous les cas, le choix du DUID utilisé par le client DHCPv6 lui incombe, mais nous ne souhaitons pas, ou ne pouvons pas intervenir sur les postes clients.

2.2 Solutions possibles

Une première solution à notre problème consiste à forcer dynamiquement la valeur du DUID d'origine, quel qu'il soit, par un DUID-UUID ou l'UUID n'est ni plus ni moins que la MAC de l'interface réseau émettrice de la requête. Cela demande de maintenir une base de correspondance DUID-MAC au sein du commutateur. En contrepartie cette solution est compatible avec des serveurs DHCPv6 extrêmement simple.



La seconde solution ne demande pas de maintenir une base de correspondance DUID-MAC au sein du commutateur, Elle consiste à associer dynamiquement le DUID contenu dans les messages DHCPv6 émis par le client à l'adresse MAC de l'interface réseau émettrice qui aura été renseignée au préalable dans l'annuaire LDAP sur lequel le serveur DHCPv6 va s'appuyer.



2.3 Le Software Defined Network

Les architectures traditionnelles des réseaux informatiques ne sont adaptées ni pour répondre aux exigences des entreprises d'aujourd'hui ni à celles des utilisateurs finaux qui demandent toujours plus de facilités pour accéder à de nouveaux services. Le Software Defined Network (SDN) apporte une solution aux opérateurs en changeant le paradigme d'architecture réseaux.

Dans l'architecture SDN, les plans de contrôle et de données sont découplés. L'intelligence réseau est logiquement centralisée et l'infrastructure réseau de contrôle est virtualisable. Ce changement de paradigme a pour conséquence de pouvoir disposer d'un réseau programmable, sans précédent, reposant sur un matériel ouvert et indépendant.

L' Open Networking Foundation est un consortium industriel à but non lucratif de plus de 100 membres qui mène le travail de développement du SDN et la standardisation des éléments essentiels de son architecture avec principalement le protocole OpenFlow. Celui-ci permet la communication entre les plans de contrôle centralisés et celui de données, des commutateurs réseau. Il est le premier standard conçu spécifiquement pour SDN, offrant une haute performance et un contrôle du trafic granulaire ainsi qu'une indépendance vis-à-vis des fournisseurs de commutateurs réseau.

Actuellement en cours de déploiement dans des centres de données ou sur des réseaux de campus, SDN offre au travers de contrôleurs centralisés OpenFlow, des avantages substantiels.

- L'indépendance : gestion et contrôle de commutateurs réseaux indépendants du fabricant.
- L'automatisation : à l'aide d'Interface de Programmation (API) communes, permettant de faire abstraction des spécificités des couches réseaux inférieures et de l'orchestration des systèmes et des applications d'approvisionnement de services.
- La réactivité : capacité d'offrir de nouveaux services et fonctionnalités réseaux sans avoir besoin de configurer les périphériques individuellement, ou d'attendre que le ou les fabricants des commutateurs les développent.
- La fiabilité et la sécurité du réseau : politique de gestion centralisée, uniforme et automatisée des périphériques réseaux entraînant moins d'erreurs de configuration.
- Le service : application de politiques de services lors de la session de l'utilisateur, en fonction de ses périphériques et du niveau de ses demandes
- L'expérience utilisateur : les applications utilisateurs communiquent avec les applications réseaux.

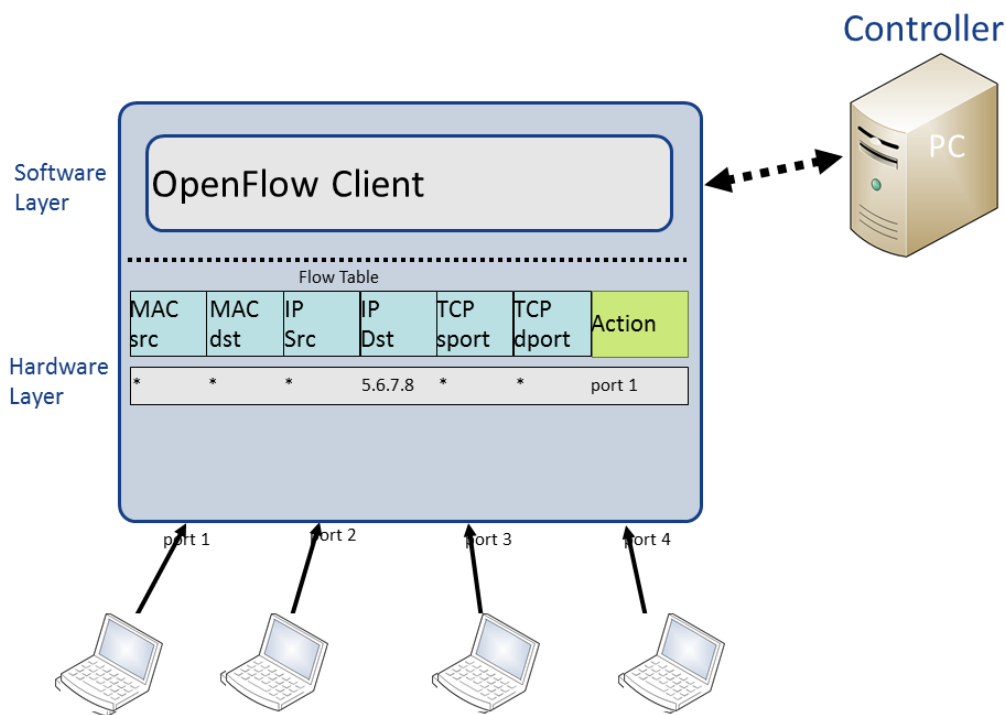
SDN propose une architecture réseau dynamique et flexible qui protège les investissements et favorise l'innovation. Les réseaux qui étaient statiques peuvent évoluer vers une plate-forme de prestation de services extensible, capable de répondre rapidement à la demande d'évolution de l'utilisateur final.

2.4 Le protocole OpenFlow

- Comment fonctionne OpenFlow ?

Dans un routeur ou un commutateur classique, la transmission de paquets (plan de données) et les décisions de routage (plan de contrôle) sont exécutées sur le même dispositif. Un commutateur OpenFlow sépare ces deux fonctions. Le plan de données réside encore sur le commutateur, tandis que les décisions de routage de haut niveau sont déplacées vers un dispositif de commande séparé, généralement un serveur standard « le contrôleur OpenFlow ». Le commutateur et son contrôleur communiquent via le protocole OpenFlow, qui définit les messages de contrôle, tels que des paquets reçus, le paquet à commuter, la modification de la table de commutation, et l'obtention des statistiques du commutateur.

Le plan de données d'un commutateur OpenFlow présente une table d'abstraction de flux (ou flow table); chaque entrée du tableau de flux contient un ensemble de champs des paquet et une action (par exemple, commuter ces paquets, sans modification des champ, ou drop) . Quand un commutateur OpenFlow reçoit un paquet qu'il n'a jamais vu avant, pour lequel il n'a aucune entrée équivalente dans la table des flux, il envoie ce paquet au contrôleur. Le contrôleur prend alors une décision sur la façon de traiter ce paquet, suppression ou ajout d'une entrée dans la table de flux, et sur la façon de transmettre les paquets similaires à l'avenir.



2.5 Dell Switch Split-Data-Plane

Pour pousser plus loin les fonctionnalités matériels des commutateurs Ethernet, DELL Research a mis au point une nouvelle architecture ayant un second plan de données pour faciliter le développement de projets innovants.

Cette architecture permet, en utilisant un Processeur Réseau (Network Processor Unit), de supporter de multiples extensions et traitements des flux contrôlés par OpenFlow (Chiffrement, compression, accélération TCP, etc)

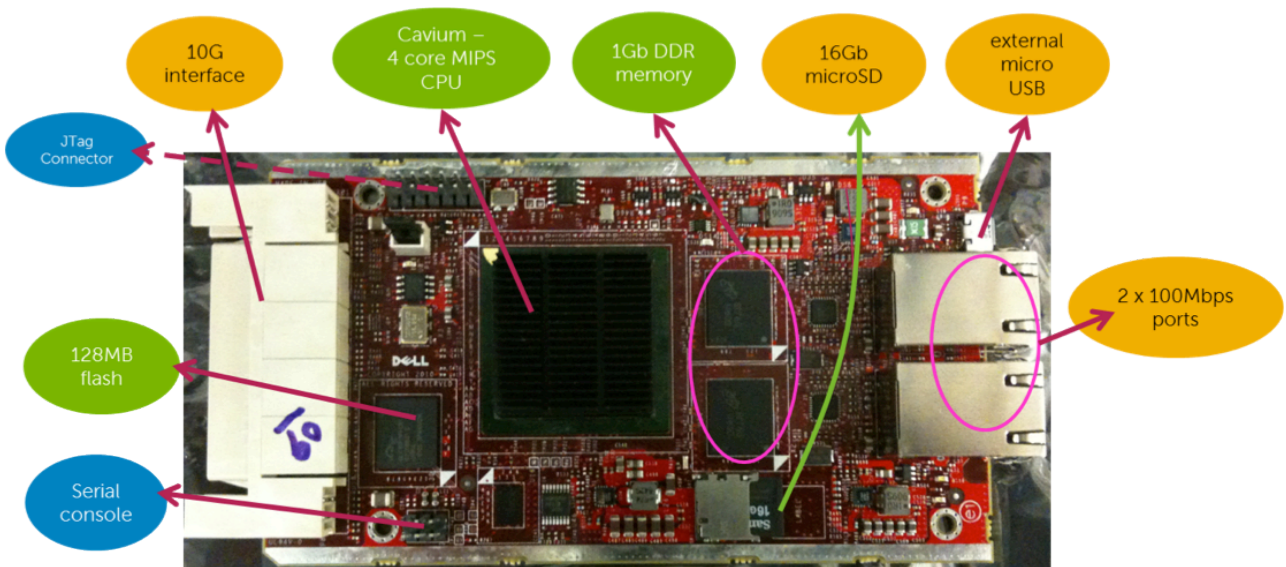
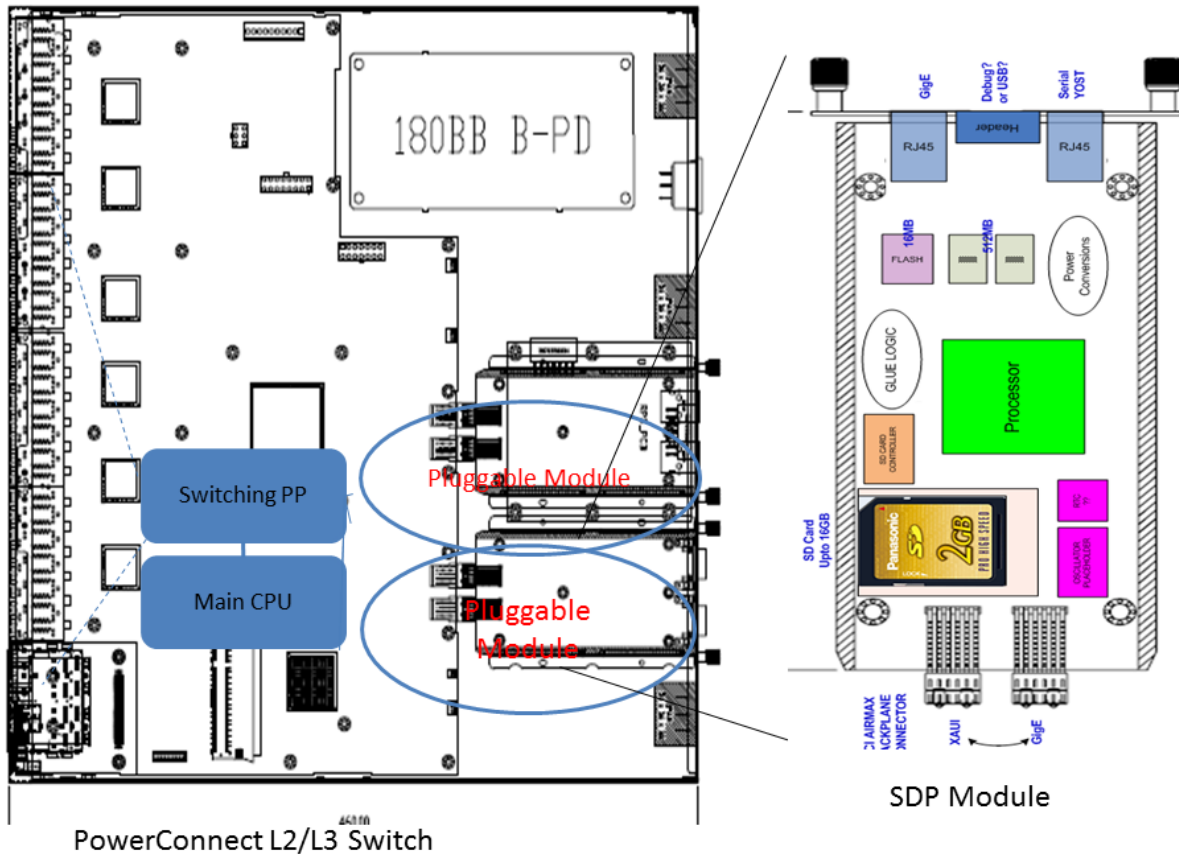
Il est possible d'aller au delà de l'architecture OpenFlow SDN et de développer en C/C++ ou en Python directement sur le NPU.

L'architecture Split-Data-Plane (SDP) tire parti d'un constat simple. Sur de nombreux scénarios de déploiement, la grande majorité du trafic ne nécessite pas de traitement et peut être gérés par des commutateurs avec un processeur de commutation sans extension. Les micro-flux nécessitant un traitement peuvent être traitées par un second plan de données, programmable via logiciel.

2.5.1 Architecture matérielle

DELL a fait le choix du processeur réseau Octeon CN5230. Celui-ci possède 4 cœurs MIPS 64 cadencés à 750MHz avec des sections d'accélération matérielle spécialisées pour le traitement des flux TCP, l'encryptions des données et la qualité de service. Le CN5230 est capable de traiter jusqu'à 6 milliards d'instructions par seconde, ce qui permet de traiter jusqu'à 4 Gbps de trafic bidirectionnel. Le NPU fait tourner un noyau Linux Debian 2.6 et un dépôt de packages est maintenu par Cavium..

Le langage de développement principal est le C, Dell fournissant un *Advanced Framework* avec plusieurs exemples et un outil qui permet d'envoyer les applications directement sur le network processor pour exécutions.



2.5.2 SDK et Framework

Le SDK de base est fourni par Cavium. C'est grâce à lui qu'il est possible de cross-compiler les programmes en C pour le NPU MIPS sur un PC x86. La distribution supportée par Cavium pour la compilation est une Fedora 16.

Le SDK contient un ensemble d'exemples d'applications comme le cryptage, la compression et un Proxy ICMP. L'un des objectifs est de fournir aux nouveaux développeurs un point de référence pour la création de leurs nouvelles applications.

L'ensemble des sources et du framework est accessible à partir de l'URL : <http://www.xflowresearch.com/dellspd>

2.5.3 Policy Based Routing

- Le Split-Data-Plane est reconnu comme une interface 10G, interne au commutateur. Le trafic n'est pas commuté naturellement vers cette interface. Pour que les flux lui soient dirigés depuis le commutateur il faut appliquer des règles de Policy-Based-Routing fondées sur des Access Lists qui permettent de choisir plus finement quels paquets seront redirigés vers le SDP.

3 Conclusion

Le portage du code de spoofing DHCPv6 est actuellement en cours au sein du Split-Data-Plane. A la vue des possibilités qu'il apporte, il ne fait aucun doute que cette architecture offre des perspectives extrêmement intéressantes et dont les administrateurs réseau n'ont pu que rêver pendant de longues années. Le SDP ne se limite pas uniquement aux applications SDN; il est possible de programmer des applications réseaux spécifiques, qui seront supportées entièrement par le commutateur. Cela ouvre la porte à une nouvelle dimension dans le domaine du réseau.