# EXTENSION DE LAN

Arnaud FENIOUX (@afenioux) – France-IX – afenioux@franceix.net
Fabien VINCENT (@beufanet) – OVH – fabien.vincent@corp.ovh.com
Stefano SECCI – LIP6 – stefano.secci@lip6.fr

**mots-clés :** ***RÉSEAU / ETHERNET / ENCAPSULATION L2 / MPLS / VPLS / VXLAN / EVPN / MTU / SINGLE DOMAIN OF FAILURE / POINT-TO-POINT / MULTIPOINT-TO-MULTIPOINT***

*Nous présentons dans cet article les solutions techniques permettant d'étendre un LAN sur plusieurs sites distants. Il existe en effet plusieurs protocoles ayant des buts ou moyens différents pour réaliser cela.*

## 1 La problématique du niveau 2

Ethernet commença au temps du coaxial, sur un média partagé où toute machine connectée à ce segment pouvait communiquer directement avec les autres, mais à tour de rôle, car elles étaient dans un seul domaine de collision. Le coaxial fut rapidement remplacé par une paire de cuivre torsadée et une topologie en étoile avec un hub au centre, permettant de garder les mêmes mécanismes et protocoles.

Pour interconnecter plusieurs segments Ethernet et éviter des domaines de broadcast trop importants, il fallait donc impérativement utiliser un routeur de niveau 3 **[I]**. By design, Ethernet n'avait pas besoin des évolutions apportées par le monde IP :

- absence de champ TTL ;
- manque d'outils de troubleshooting (OAM) ;
- absence de signalisation en cas d'échec de transmission (si une trame corrompue - ou de mauvaise taille - est reçue, elle est supprimée silencieusement).

L'apparition des switchs permit de scinder le domaine de collision, mais de conserver un unique domaine de broadcast, et donc d'interconnecter plus de machines, plus facilement, plus vite.

Le protocole STP (*Spanning Tree Protocol*) fut inventé dans le but de prévenir les risques de boucle **[2]** et d'avoir plusieurs liens de backup pour relier deux segments, tout en conservant une compatibilité avec l'historique. L'essor d'Ethernet dans les années 2000 entraîna un développement de cette technologie au-delà du LAN et chaque constructeur tenta d'imposer sa propre solution propriétaire pour améliorer STP.

Les besoins des utilisateurs furent d'abord l'interconnexion de réseaux IP distants (résolu grâce au transport IP). Puis, dans un second temps, le cloisonnement de leurs réseaux devint important (par exemple une entreprise ayant un siège et plusieurs bureaux éloignés). Ce besoin fut résolu d'abord grâce aux interconnexions point à point avec une architecture centralisée (en hub & spoke), puis par L3VPN. Dans le cas du L3VPN, une partie du routage est déportée dans le réseau de l'opérateur (ce qui ajoute une contrainte de gestion pour le client, mais facilite le troubleshooting).

Il apparut par la suite le besoin d'étendre le LAN de manière décentralisée sur plusieurs datacenters et de permettre la migration de VM afin d'assurer une continuité de service.

## 2 Les différents types d'encapsulations

Tous les types d'encapsulations présentés dans cet article ne garantissent pas la confidentialité des données transitant par le backbone de l'opérateur. Un client utilisant ces services devrait donc implémenter une couche de cryptographie s'il ne fait pas confiance au réseau de son opérateur.

### 2.1 Dot1q et 802.1ad (QinQ)

Même si on ne peut pas parler d'encapsulation à proprement parler, la solution la plus économique pour relier plusieurs sites (ou PoPs : *Point of Presence*) est d'utiliser un lien de niveau 2, généralement configuré en trunk afin de faire transiter plusieurs VLAN.

La norme 802.1q (souvent appelée Dot1q) rajoute un champ de 32 bits entre l'adresse MAC source et l'Ethertype original, afin de taguer les trames Ethernet.



Les 16 premiers bits sont utilisés pour le TPID (*Tag Protocol Identifier*) - à la place de l'EtherType - et sont positionnés à 0x8100 afin de savoir qu'il s'agit d'un tag. S'en suivent un champ PCP (*Priority Code Point*) de 3 bits pour la priorité et un champ DEI (*Drop Eligible Indicator*) de 1 bit pour autoriser les trames avec ce champ à 1 à être supprimées en priorité en cas de congestion. Le champ VLAN ID, pour finir, est codé sur 12 bits, ce qui ne permet que 4094 réseaux virtuels distincts (0x000 et 0xfff sont réservés).

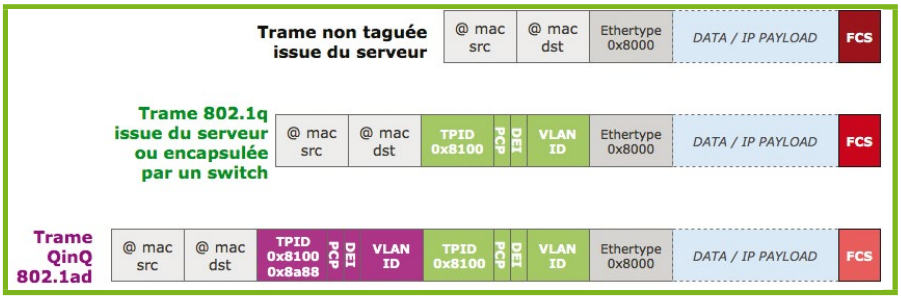


*Figure 1 : Ajout des tags 802.1q et 802.1ad à une trame Ethernet.*

L'utilisation de 802.1ad (souvent appelé QinQ) permet d'ajouter un second tag sur une trame éventuellement déjà taguée, cela afin de ne pas mélanger les VLAN des clients de ceux de l'opérateur. L'inner tag (s'il est présent) est réservé à l'usage du client (TPID à 0x8100), alors que l'outer tag est utilisé dans l'infrastructure de l'opérateur (TPID normalement à 0x88a8, mais il peut être à 0x8100 ou 0x9100 pour de vieilles implémentations **[3]**). En théorie seulement deux tags sont autorisés, mais en pratique on peut rencontrer des trames avec plus de deux tags. L'ajout de chaque nouveau tag allongera la trame Ethernet de 4 octets, il conviendra donc de veiller à configurer une MTU suffisamment élevée dans le backbone de l'opérateur afin d'éviter le drop des trames les plus grandes.

QinQ n'offre pas une réelle séparation entre le domaine du client et celui de l'opérateur (par exemple : gestion des multiples variantes de STP dans le réseau du client) et doit plutôt être vu comme un moyen de contourner le nombre restreint de VLAN disponibles.

La sécurisation des liens et la désactivation des chemins de backup se fait le plus généralement grâce à STP, ou manuellement. Entraînant une non-utilisation d'une partie de la capacité du backbone de l'opérateur ainsi que l'apparition éventuelle de chemins sub-optimum.

Ces limitations, ainsi que le besoin d'étendre le LAN alors qu'IP était déjà déployé, ont poussé l'industrie à se tourner vers d'autres solutions d'overlay, permettant entre autres le multipath forwarding et le load balancing.

## 2.2 MPLS / VPLS

VPLS (*Virtual Private LAN Service*) est une technologie de L2VPN permettant d'interconnecter plusieurs segments Ethernet distants afin de créer un seul domaine de broadcast à travers un réseau qui repose en pratique sur IP/MPLS. Nous allons donc commencer par expliquer le fonctionnement de MPLS.

### 2.2.1 EoMPLS : Ethernet over MPLS

Les routeurs MPLS (*Multi-Protocol Label Switching*) sont bien plus coûteux que de simples switchs Ethernet, mais permettent la transmission de trames de n'importe quel protocole, par exemple Ethernet, à travers le réseau d'un opérateur tout en profitant de la flexibilité et des avantages des protocoles de routage IP.

Il existe deux protocoles de signalisation pour l'échange de labels entre les routeurs MPLS. Le premier, LDP (*Label Distribution Protocol*) est le plus simple et se repose sur le protocole de routage interne (IS-IS ou OSPF) pour choisir le meilleur chemin vers une destination.

Le second, RSVP-TE (*Resource Reservation Protocol - Traffic Engineering*) permet de gérer plus finement l'utilisation de chacun des liens. Il permet également, grâce à FRR (*MPLS Fast Reroute*) **[5]** d'avoir un chemin alternatif, pré-calculé et de basculer le trafic sur ce chemin en moins de 50ms en cas de panne. Finalement, MPLS OAM **[6]** propose plusieurs outils pour l'administration et le troubleshooting tels que MAC ping, MAC traceroute, LSP ping, etc.

Il est courant que le réseau client, ainsi que le backbone de l'opérateur soient basés sur Ethernet, on se retrouve alors avec une encapsulation « Ethernet over MPLS over Ethernet ». Les trames MPLS ont alors un EtherType fixé à 0x8847 pour l'unicast (et 0x8848 pour le multicast) (Figure 2).

Chaque paquet Ethernet entrant dans un réseau MPLS se voit tagué avec deux labels, le « tunnel label » (label extérieur) est utilisé pour le transport dans le backbone, alors que le VC label (label intérieur) est utilisé pour indiquer l'interface de sortie du PE (*Provider Edge* : routeur qui est connecté à l'équipement du client).

**Note**

**Nous ne présentons dans ce dossier que des solutions techniques permettant de relier plusieurs sites en multipoint-to-multipoint, les solutions point-to-point telles que L2TP, GRE, IPSec, OpenVPN, EtherIP, etc. ne sont dont pas évoquées ici (exception faite pour MPLS). De plus, nous ne prétendons pas être exhaustifs, d'autres solutions existent, telles que PBB (*Provider Backbone Bridge*), NVGRE (*Network Virtualization Using Generic Routing Encapsulation*) ou STT (*Stateless Transport Tunneling*), voir [4].**

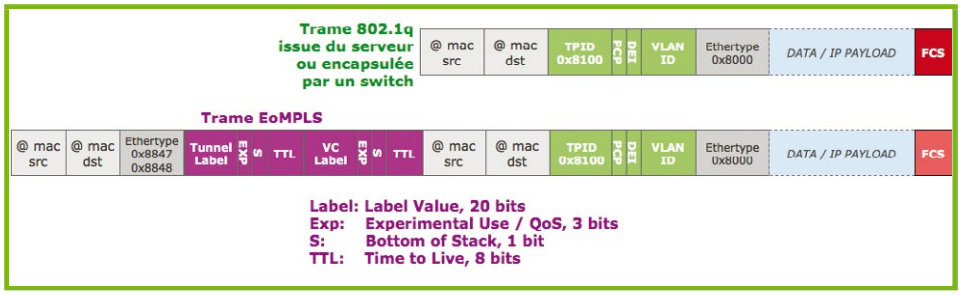


*Figure 2 : Trame EoMPLS over Ethernet.*

Cette trame MPLS traverse ensuite le réseau en suivant la LSP (*Label Switched Path*) définie par le protocole de signalisation. Elle est forwardée par les routeurs uniquement en fonction du label extérieur (qui est swappé à chaque étape). Cette opération est bien moins coûteuse qu'un lookup de la table de routage.

Les LSP sont unidirectionnels, il faut donc qu'ils soient établis dans les deux sens entre les deux routeurs d'extrémités afin qu'il puisse y avoir une communication bidirectionnelle.

### 2.2.2 VPLS

En VPLS, du point de vue du CE (*Customer Edge* : routeur du client), le réseau de l'opérateur se comporte comme un switch virtuel, cette fonctionnalité est donc très appréciée par les IXP (Points d'échanges Internet) qui sont souvent sur plusieurs sites. Une connectivité en full-mesh est donc nécessaire entre tous les sites afin que chaque routeur PE puisse apprendre les adresses MAC sur ses ports et pseudo-wires (PW), dupliquer les paquets Broadcast, Multicast et flooder le trafic Unknown Unicast (on parle de trafic BUM).

Tous les sites sont reliés entre eux en full-mesh grâce à l'utilisation de liens pseudo-wires (chaque site n'a pas besoin d'être physiquement relié avec tous les autres en direct). Afin d'éviter tout risque de boucle, le principe de « split horizon » est utilisé : le trafic BUM reçu via un PW ne doit pas être forwardé à un autre PE.

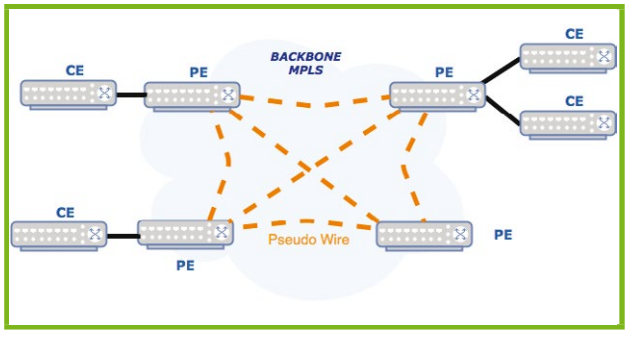


*Figure 3 : Exemple de LSP dans un VPLS.*

Un avantage d'utiliser des PW est qu'en cas de rupture du data-plane (panne d'un lien ou d'un équipement dans le cœur du réseau), le trafic sera automatiquement re-routé de manière transparente via un chemin alternatif dans le réseau de l'opérateur. Le calcul du nouveau chemin repose sur l'IGP utilisé dans le réseau de l'opérateur.

Ces tunnels point-à-point sont généralement créés en utilisant une encapsulation EoMPLS *(Ethernet over MPLS*), mais pourraient être établis grâce à L2TPv3 ou GRE. Dans le cas de tunnels MPLS, il existe deux méthodes pour établir ces PW en full-mesh, soit via BGP (*Border Gateway Protocol*) décrit dans la RFC 4761, soit via LDP (*Label Distribution Protocol*) comme décrit par la RFC 4762. Bien qu'étant nommées VPLS, ces deux méthodes sont incompatibles entre elles.

Les paquets MPLS VPLS ont deux labels : le label extérieur est utilisé pour la commutation normale entre les routeurs MPLS, alors que le label intérieur est utilisé pour dissocier les instances VPLS (dans le cas où il y en aurait plusieurs). Il est possible de transporter un seul, ou plusieurs VLAN, dans une instance VPLS.

Le client est libre d'utiliser n'importe quel protocole de niveau 2 (y compris STP), toutes les trames étant encapsulées par MPLS.

Une des faiblesses de VPLS est que l'apprentissage des adresses MAC de tous les équipements des clients se faisant via le data-plane (comme pour un switch traditionnel) cette technique (*flood & learn*) fait donc énormément appel aux paquets floodés et broadcastés, paquets qui doivent être dupliqués et transmis par le premier PE.

Ce problème est en passe d'être résolu par EVPN. Ethernet VPN est un overlay sur réseau MPLS ou VxLAN qui, contrairement à VPLS, utilise le control-plane afin d'apprendre les adresses MAC. L'annonce et l'apprentissage des adresses MAC dans le control-plane s'effectuent grâce à une nouvelle famille d'adresses dans MP-BGP (*Multi-Protocol BGP*) et de nouvelles communautés BGP étendues. Ce protocole permet de gérer le trafic par MAC et d'avoir par exemple deux VM sur le même site utilisant des chemins différents pour atteindre la même cible.

Nous vous invitons à regarder cette présentation du Cisco live **[7]**, qui détaille le fonctionnement de MPLS VPLS et EVPN.

## 2.3 VxLAN

### 2.3.1 Qu'est-ce que VxLAN ?

*Virtual eXtensible LAN* est un protocole de virtualisation des réseaux développé au début des années 2010 par VMware, Cisco et Arista pour pallier les besoins croissants d'isolation des machines virtuelles. Il est aisé



de comprendre le fondement de VxLAN : la norme VLAN 802.1q s'appuie sur un entête avec un identifiant codé sur 12 bits, ce qui autorise la création d'au maximum 4094 réseaux virtuels distincts VLAN. Cette limitation, jadis largement suffisante, est insuffisante aujourd'hui dans le cas des gros datacenters, les switchs « Top-of-Rack » devant absorber de plus en plus d'adresses MAC **[8]** et de VLAN, à cause des machines virtuelles.

Les technologies de virtualisation entraînent des changements conséquents dans les architectures réseaux et rendent les composants réseaux de plus en plus logiciels - OpenStack Neutron, OVS, dVS, Nexus 1000v pour ne citer qu'eux. Intégrer l'encapsulation au plus proche de l'hyperviseur permet aussi de ne pas dépendre des équipes réseau qui sont parfois moins enclines à tout automatiser. De plus, les plans de continuité ou de reprise d'activité PRA/PCA imposent de distribuer l'infrastructure sur plusieurs sites suffisamment distants. Bien souvent, ces sites ne disposent que d'une connectivité IP, et le niveau 2 est trop peu adapté à la résilience des chemins (*Spanning-Tree*), surtout en longue distance. VxLAN va répondre à ces besoins par l'usage d'une encapsulation des trames de niveau 2 dans des fragments UDP de niveau 4, en utilisant le réseau IP de niveau 3 pour étendre les domaines de broadcast.

## 2.3.2 Comment fonctionne VxLAN

### 2.3.2.1 Encapsulation de VLAN vers VxLAN

VxLAN va être utilisé en extension d'un domaine de broadcast qui est bien souvent aujourd'hui le VLAN. Lorsque ce VLAN est QinQ (Outer VLAN ou Service VLAN), on ne s'intéressera qu'au VLAN externe, celui géré par le fournisseur. L'idée étant de laisser au client la liberté de disposer des 4094 VLAN au travers de l'infrastructure QinQ (localement) + VxLAN (transport), afin qu'il puisse utiliser librement, soit une trame en VLAN natif, soit en VLAN tagué par la machine. Le VLAN externe (outer VLAN) sera lui translaté dans l'entête VxLAN par le VNI. Cet identifiant VNI, pour *VxLAN Network Identifier*, codé sur 24 bits, permet la création de 16 millions de réseaux VxLAN. Ce VNI est situé juste après les adresses IP source et destination des VTEP, pour *VxLAN Tunnel End Point*. Ces interfaces IP vont permettre d'encapsuler/décapsuler le segment UDP transmis/reçu. En fonction d'un ID de VLAN source, on translate donc celui-ci par un VNI dans une entête VxLAN (tout en conservant le inner VLAN client dans le cas d'un QinQ), puis on transmet le segment aux IPs des VTEP participant au VxLAN d'un même VNI. L'intérêt majeur est de pouvoir augmenter le nombre de domaines de broadcast à 16 millions, car les 4094 VLAN seront alors uniquement utilisés localement, derrière la VTEP. L'association est donc locale, mais pour éviter toute inconsistance, il est important de s'attacher à garder des VNIs uniques sur l'ensemble au moins du datacenter.

### 2.3.2.2 Schéma de principe

Prenons un exemple simple avec 3 VTEP, 3 VNI et autant de VLAN que de machines. Au travers du VNI 5555, nous pourrons faire communiquer les VLAN locaux 100, 340 et 250. Au travers du VNI 1000, nous pourrons faire communiquer les VLAN 122 et 210, qui sont uniques et bien différents, car pas sur le même switch ni derrière la même VTEP. Idem pour le VNI 333.

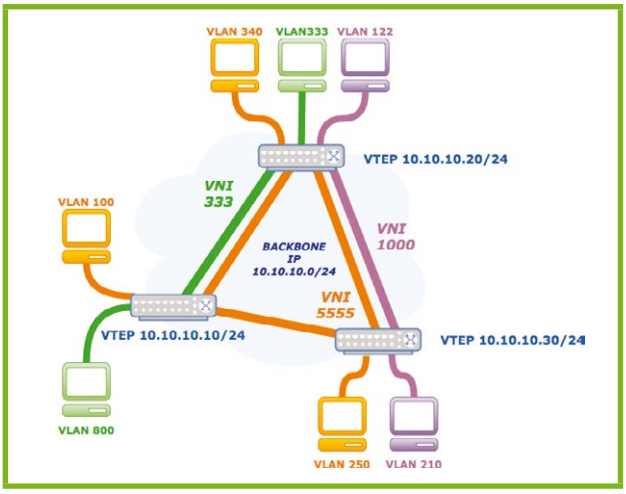


*Figure 5 : Binding des VNI / VTEP.*

## 2.3.3 Types de déploiement

VxLAN permettre d'étendre les domaines de broadcast par l'usage des réseaux IP. Cela pose un problème de taille : comment faire fonctionner des réseaux de niveau 2 travaillant uniquement sur le forwarding par adresse MAC de destination à travers des réseaux IP se basant sur des technologies de routage ?

Dans le cas où les adresses MAC sont connues, cela est simple, on forwarde le paquet encapsulé localement, ou à la VTEP identifiée comme portant l'adresse MAC. Mais comment faire lorsque l'adresse MAC n'est pas connue ? Où envoyer le paquet ? C'est le problème bien connu du « Flood and Learn », car dans les cas des paquets BUM (*Broadcast*, *Unknown Unicast*, *Multicast*), il va falloir répliquer le

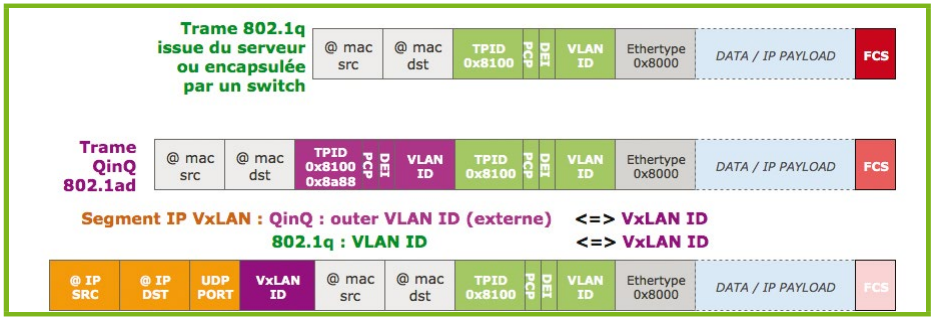


*Figure 4 : Ajout des tags VxLAN à une trame Ethernet.*

paquet sur l'ensemble des VTEP participant au réseau VxLAN pour garantir le fonctionnement normal du domaine de broadcast.

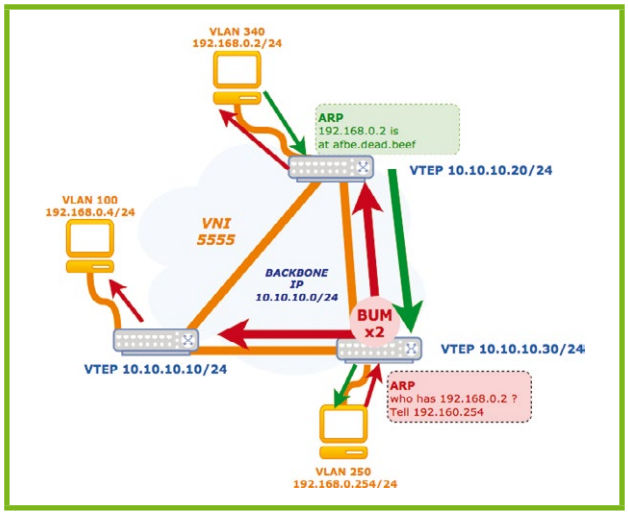


*Figure 6 : Gestion du BUM.*

Ainsi, lorsque les adresses MAC sont connues, la VTEP et son switch disposeront, en plus d'une table d'association MAC <> Port pour le forwarding, d'une table d'association MAC <> IP, l'IP étant la VTEP sur laquelle il faudra router le paquet encapsulé.

#### 2.3.3.1 Déploiement original : multicast

Par essence, le multicast était la solution la plus logique du VxLAN. C'est même celle qui a été décrite initialement dans la RFC7348. Chaque VNI constitue un groupe multicast, et un Rendezvous Point duplique les paquets vers les VTEP du groupe lorsque l'adresse MAC de destination n'est pas connue. Le retour se fait directement à l'émetteur (VTEP responsable du flood source). Cette solution est particulièrement intéressante dans le cas d'infrastructures localisées, mais le multicast ne fait pas rêver les foules, et son déploiement est souvent complexe. De plus, il est difficile d'imaginer cette solution à grande échelle, sur des liens opérateurs. Cette solution est une des premières à avoir été implémentée par Cisco, notamment sur le commutateur virtuel Nexus 1000v, utilisé pour se substituer au dvSwitch de VMware.

#### 2.3.3.2 Déploiement HER ou Unicast

Afin de pallier les problèmes du multicast, Arista et Cisco ont développé une solution alternative au multicast : le *flood and learn* avec *Head End Replication*. Dans ce cas de figure, chaque VTEP participant à un VxLAN doit connaître l'ensemble des VTEP de destination (flood-list) où les BUM doivent être transmis et répliqués. Cette solution a un avantage indéniable par sa configuration complètement décentralisée. En revanche, elle pose question sur les performances : en cas de trafic BUM important, il est possible que les CPU des équipements soient fortement sollicités au point de perdre des paquets si les cycles CPU du switch sont plus lents que l'augmentation du nombre de paquets BUM. Cette limitation commence à être contournée par l'implémentation du HER en hardware pour éviter de solliciter le CPU inutilement.

Un autre problème de cette solution est que la configuration se complexifie de manière exponentielle dès que le nombre d'associations VLAN-VxLAN augmente significativement et encore plus lorsque le nombre de VTEP augmente. Il faut donc également tenir compte des limitations hardware des switchs dont les tables sont souvent limitées en taille, ce qui peut être rapidement un frein ou une difficulté au déploiement de la méthode Unicast. La « rogue » VTEP est également un point noir de cette méthode de déploiement. À la réception du paquet, il n'est pas possible de savoir si la trame est encapsulée par une VTEP ayant légitimité à le faire. La décentralisation engendre de fait un risque plus important de voir apparaître ce type de comportement, comme dans le cas des réseaux sans fil.

#### 2.3.3.3 Déploiement MP-BGP / E-VPN

Ces deux méthodes n'étant pas ultimes ou ayant leurs propres limitations, l'idée est venue de gérer les réseaux VxLAN avec des extensions déjà existantes dans le protocole BGP, MP-BGP (*Multi Protocol BGP*). Ceci permet de séparer le Control Plane (décision de routage/forwarding) et le Data Plane (actions de routage/forwarding). Dans le cas de MP-BGP, on remplace le RendezVousPoint par un ou plusieurs Route Reflector qui occupera le rôle de Control Plane. Ce dernier va s'occuper pour la famille d'adresses E-VPN de maintenir la table de routage des MAC au travers du VxLAN et de la distribuer par le protocole BGP aux switchs clients portant chacun une VTEP, tout en conservant HER, mais en simplifiant la configuration. Ces switchs seront alors uniquement des membres du Data Plane.

Cette méthode **[9]** de déploiement est très intéressante, car au-delà d'utiliser des technologies existantes et éprouvées (BGP), celle-ci introduit des mécanismes de NLRI (*Network Layer Reachability Information*) permettant de limiter les floods de BUM, voire de programmer des routes vers certaines MAC en dur pour sécuriser l'infrastructure finale. Cette méthode permet également de sécuriser plus facilement les VTEP, par l'ajout de mécanismes d'authentification déjà existants dans BGP.

### 2.3.4 Impacts

#### 2.3.4.1 Impacts sur la MTU

Ethernet a une MTU de 1500 ou de 9000 octets dans le cas des JumboFrames. L'entête VxLAN rajoute 50 octets à la trame Ethernet (14 Ethernet + 8 UDP + 20 IP + 8 VxLAN), il est donc conseillé d'augmenter la MTU du réseau de transport de 50 à 100 octets pour d'éventuels besoins futurs ou pour QinQ. Cela peut engendrer des complications sur les protocoles de routage, souvent sensibles à des changements de la MTU.



#### 2.3.4.2 Impacts sur l'ECMP

Les protocoles de routage interne ou les agrégats de type LACP répartissent la charge sur plusieurs liens en fonction de hashs suivant des combinaisons définies (MAC/IP/Port Source). Le transport de trames L2 dans des segments UDP peut poser des problèmes engendrant un déséquilibrage de la charge. Il est important de veiller et contrôler auparavant la méthode utilisée afin d'éviter des déséquilibres qui pourraient s'avérer fatals pour l'infrastructure de transport.

#### 2.3.4.3 Contrôles des trames

VxLAN ajoute un entête de transport, et utilise la correction d'erreur FCS du paquet original. Ainsi il est impossible de savoir, lorsque le paquet reçu est non conforme, si l'erreur provient du réseau de transport ou du domaine de broadcast. Si le réseau de transport est géré par un tiers, cela peut devenir très compliqué de trouver le chemin générant des CRC, le segment VxLAN pouvant passer par plusieurs chemins IP différents en fonction des algorithmes de hachage.

#### 2.3.4.4 Charge des switchs

La déduplication des trames BUM impacte fondamentalement les équipements, qui doivent traiter des paquets qui ne leur sont pas destinés en encapsulation. Lorsque cette réplication est réalisée en software par le CPU, il est évident que des pertes de trames BUM sont à prévoir si surcharge. Les protocoles multicast, comme VRRP, qui pourraient être utilisés dans le VxLAN sont alors directement impactés.

#### 2.3.4.5 Impact sur la sécurité

Les VxLAN cassent le modèle des VLAN, souvent terminés sur des firewalls. Dans ce cas, il faut bien prendre en compte que le domaine de broadcast qui pourra être étendu en multi-sites n'est pas chiffré et peut utiliser des liens d'un réseau IP pas toujours maîtrisé par le client. Le VxLAN routing, qui permet de router un VxLAN dans un autre, augmente encore ce risque de casser la sécurité des VLAN.

## 2.4 De l'encapsulation au routage Ethernet (TRILL/SPB)

En parallèle de MPLS et ses généralisations, d'autres protocoles d'encapsulation purement L2 ont été spécifiés et commercialisés dans la dernière décennie, avec comme objectif d'augmenter les performances et d'ajouter des fonctionnalités pour la virtualisation des réseaux, tout en dépassant les limitations de STP.

Il vaut la peine de mentionner TRILL (*Transparent Interconnection of a Lot of Links*) et SPB (*Shortest Path Bridging*). Ces deux protocoles se ressemblent dans la mesure où ils mettent en place une encapsulation L2-L2, font appel au protocole de routage ISIS, et utilisent un TTL dans la trame pour palier aux boucles.

Toutefois SPB et TRILL diffèrent fondamentalement dans le mode d'opération et de déploiement :

- SPB, standardisé par l'IEEE avec 802.1aq, fait plus directement appel à 802.1ah et PBB et leurs fonctions de OAM, etc.
- TRILL, standardisé par l'IETF avec différents RFCs, a au contraire adopté une approche type '*tabula rasa*'.

Alors que SPB requiert un déploiement ubiquitaire sur tous les switchs du réseau, TRILL permet d'avoir des segments multi-sauts en Ethernet entre deux nœuds TRILL, appelés *Router-Bridges* (RBs). Cela permet de considérer TRILL comme un protocole d'extension de LAN, à condition d'utiliser une liaison de niveau 2 pour relier une infra de DC géographiquement distribuée. Avec SPB : une trame entrante est encapsulée avec pour adresse MAC destination le switch de sortie du réseau de backbone SPB, alors que dans TRILL la MAC de destination est l'adresse du prochain saut TRILL-ISIS sur le plus court chemin vers le RB de sortie (et l'adresse MAC source est celle du RB générant la trame TRILL).

De la sorte, TRILL se présente comme un protocole apte à pouvoir gérer la mobilité des machines virtuelles (ou VMs), car sa méthode d'encapsulation permet d'ajouter une information sur la localisation d'une MAC sans avoir besoin de mettre à jour tous les switchs de l'infrastructure comme dans SPB. Il manque cependant à TRILL l'identification VNI pour le considérer comme un vrai protocole de virtualisation de réseau. Toutefois, cela a été exploré et implémenté par Gandi, voir **[10]**.

Nous avons présenté TRILL très synthétiquement. Toutefois, la principale limitation de TRILL est qu'il nécessite d'utiliser des liens L2 pour relier plusieurs sites. Une encapsulation TRILL sur UDP/IP **[11]** est discutée, mais cela pourrait laisser perplexe face à des solutions existantes telles que VxLAN, ou LISP.

## 2.5 Qu'est-ce que LISP ?

Le *Locator/Identifier Separation Protocol* (LISP) est un protocole qui peut être utilisé pour la virtualisation des réseaux. Il a été développé à l'IETF avec comme domaine d'application primaire le routage Internet (diminution de la taille des tables de routage BGP), mais il a été utilisé dans différents contextes comme, par exemple, dans les réseaux de datacenters pour la gestion de la mobilité des machines virtuelles. Cisco est l'industriel qui s'est investi le plus dans la standardisation de LISP et de son intégration dans des produits commerciaux, à ce jour.

Le principe de LISP est d'effectuer une encapsulation IP sur IP, où les adresses IP externes - appelées RLOCs (*routing locators*) - indiquent le localisateur de la source et de la destination (le serveur où se trouve une machine virtuelle, par exemple), et les adresses IP

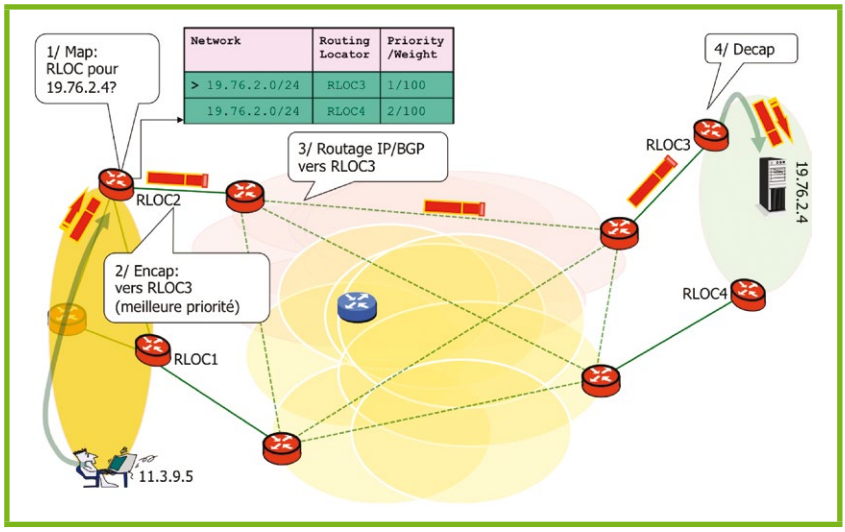


*Figure 7 : Routage LISP : aller avec mapping entry déjà dans le cache.*

internes – appelées EIDs (*End-point Identifiers*) indiquent les terminaux. Pour cela, LISP s'appuie sur un plan de contrôle externalisé, et accédé en mode « pull » : on demande au fur et à mesure des besoins les informations de localisation, quand le trafic vers une destination IP dont on ne connaît pas la localisation, arrive. Lorsqu'un routeur LISP reçoit du trafic vers une destination IP qui est inconnue du plan de contrôle LISP (trafic non LISP), le message est transféré tel quel sans encapsulation LISP.

Un exemple de communication Map&Encap LISP est donné dans les figures suivantes. Dans la figure 7, il s'agit d'un paquet allant de la machine 11.3.9.5 à la machine 19.76.2.4. Dans la figure 8, il s'agit du chemin de retour. On suppose pour l'aller, que le routeur LISP de sortie (sur le chemin de sortie) possède déjà l'information sur la localisation dans son mapping cache ; 19.76.2.4 a deux RLOCs, dont un prioritaire: une métrique de priorité est prévue à cet effet. On suppose maintenant que pour le retour le routeur LISP de sortie ne possède pas de mapping pour 11.3.9.5. C'est dans ce cas que le plan de contrôle rentre en jeu, en interrogeant avec un message map-request le système de mapping, qui lui répondra avec un map-reply. Dès le mapping installé dans la mapping cache, le paquet est encapsulé vers RLOC1. À noter que cette fois les priorités pour les deux RLOCs de destination sont les mêmes : la seconde métrique, le « weight », entre en jeu en donnant les taux de partage de charge sur les RLOCs.

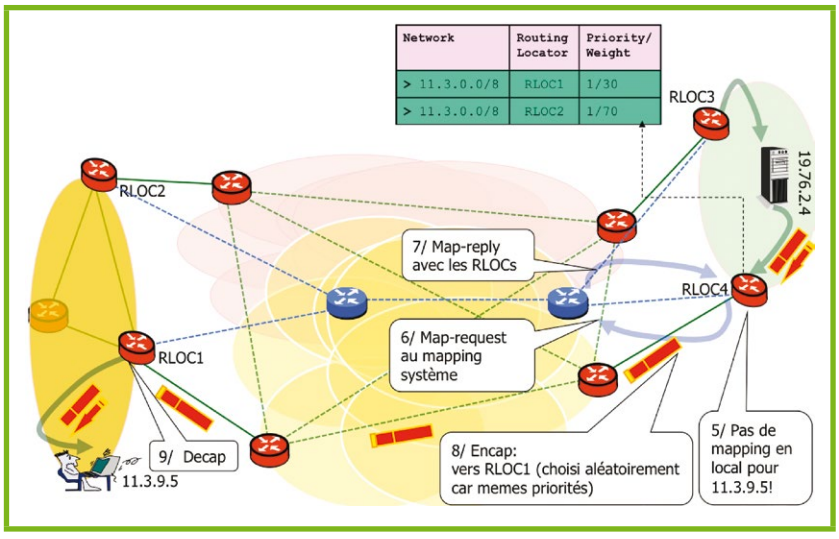


*Figure 8 : Routage LISP : retour avec requête au mapping system.*

En effet, le plan de contrôle inclut un système de mapping qui garde les associations IP EID<>IP RLOC, de façon duale à ce fait par DNS pour les associations IP<>nom, par exemple. À la différence de certains de ses prédécesseurs (qui effectuent la séparation entre l'identifiant et le localisateur IP comme HIP et SHIM6), LISP n'a pas vocation à modifier les nœuds EID : la logique LISP est implémentée uniquement sur des routeurs IP qui se trouvent sur le chemin du trafic de bordure (passerelle par défaut ou routeur de bordure d'un réseau local).

### 2.5.1 Encapsulation LISP

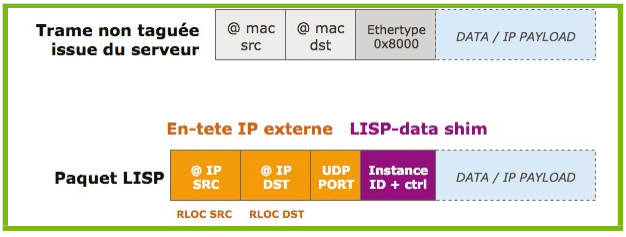


*Figure 9 : Ajout du shim LISP.*

La trame venant du terminal peut éventuellement être taguée ou traverser des segments QinQ ou d'autres technologies de niveau 2. Lorsqu'elle atteindra un routeur LISP (son RLOC), le routeur LISP s'intéressera seulement au paquet IP, qu'il encapsulera dans un autre paquet IP vers le RLOC de la destination (donné par le plan de contrôle) en utilisant UDP avec un port de destination LISP-data. Un shim (champ intermédiaire) de 8 octets est ajouté entre l'entête UDP et le paquet client.

Dans le shim LISP, les 4 derniers octets peuvent être utilisés pour désigner une 'INSTANCE-ID', qui peut faire office de VLAN étendu ou identifiant de réseau virtuel (VNI) comme dans VxLAN. Cet usage particulier n'est pas explicité dans le standard, mais utilisé de-facto par certains. En effet, si on compare le paquet VxLAN au paquet LISP, la différence est minime : les deux utilisent



une encapsulation IP, UDP, un shim de 8 octets, mais VxLAN transporte une trame Ethernet alors que LISP transporte un paquet IP. Les similitudes entre VxLAN et LISP ne s'arrêtent pas là : dans une certaine mesure le VTEP est l'équivalent du RLOC. Une grande différence entre les deux existe toutefois, au niveau du plan de contrôle, totalement découplé dans LISP, sur lequel ne nous attarderons pas, mais nous vous invitons à le découvrir dans **[12]** (ainsi que son implémentation effectuée par le LIP6 **[13]**).

### 2.5.2 Gestion de la mobilité des machines virtuelles

Il est légitime de se demander comment LISP peut servir pour étendre un LAN de niveau 2 à partir du moment où le terminal et son RLOC sont supposés communiquer en L3. Plutôt que d'effectuer de la segmentation des réseaux virtuels par VNI/INSTANCE-ID, à ce jour l'utilisation principale de LISP dans les DCs est le pilotage du trafic et l'optimisation du routage vers les machines virtuelles (VM). Cet usage a intéressé Cisco, mais aussi VMWare et EMC parmi d'autres **[14-16]**.

Quel est donc le principe de LISP pour la mobilité des VMs ? L'idée est d'émuler le même LAN dans les différents endroits où la VM pourrait être migrée. Lorsque la VM est migrée, elle garde son adresse IP, son adresse MAC ainsi que sa table de routage IP et sa table ARP. Pour avoir un fonctionnement correct, tous les next-hops dans sa table de routage IP doivent être visibles et opérationnels dans les différents endroits où la VM peut être migrée, mais aussi avec les mêmes adresses MAC. Il s'agit de répliquer le LAN vu par la VM derrière les clusters de virtualisation où elle pourrait être migrée. Il est donc sage de rendre sa table de routage la plus compacte possible, avec très peu de routes, et avec la route par défaut pointant vers le routeur LISP qui aura la même adresse MAC sur les différents sites, mais un RLOC différent.

Une VM sera donc migrée d'un DC à un autre DC, ou d'une salle à une autre salle, sans avoir besoin que les différents sites soient interconnectés en L2. Après la migration, le plan de contrôle doit faire le nécessaire pour associer l'IP de la VM à sa nouvelle localisation (RLOC). Pour cela, deux façons principales ont été proposées :

- La première, par Cisco, consiste à préconfigurer au niveau des routeurs LISP les adresses IPs qui sont mobiles : quand le routeur LISP reçoit des données d'une telle IP et que le RLOC actuellement associé à cette IP par le plan de contrôle n'est pas le sien, il sollicitera au plan de contrôle une mise à jour du RLOC pour « s'approprier » de la fonction de localisation de la VM. Une telle approche a le mérite d'être assez légère, mais si elle n'est pas complétée par des mécanismes de sécurisation, pourrait être dangereuse.
- La seconde est proposée dans **[17]** et s'appuie sur un message de plan de contrôle (avec authentification HMAC) de l'hyperviseur vers le routeur LISP (ou directement le système de mapping IP<>RLOC) dès réception de la VM. Ainsi, dans **[6]** il est démontré que la migration à chaud de VMs à travers l'Internet sur de très longues distances peut se faire avec un temps d'interruption inférieur à la seconde, et donc sans interrompre les connexions TCP. La migration à chaud n'est évidemment pas la seule façon de migrer un VM. Snapshotting et duplication de VMs pilotées avec LISP sont d'autres techniques envisagées pour garantir la résilience en cas de désastre de grande envergure, voir **[14-16]**.

Si on s'intéresse au modèle de déploiement avec le moins d'impact sur l'infra réseau, avec une VM qui n'a que sa passerelle par défaut dans sa table de routage, alors la VM et son routeur LISP - dans le jargon nommé xTR, pour *Ingress/Egress Tunnelling Router* - peuvent être co-localisés dans le même serveur de virtualisation. Chaque serveur de virtualisation aura son propre xTR sur une VM (par exemple en utilisant **[13]**), qui sera la passerelle par défaut de la VM qui est migrée.

Alternativement à la mise en VM du xTR, pour les fonctionnalités du plan de transfert LISP, on peut s'appuyer sur OpenVswitch, qui supporte depuis quelques années l'encapsulation LISP. Dans ce cas, OVS est piloté à distance, notamment avec OpenDayLight, voir **[18]**, ou d'autres contrôleurs comme par exemple OpenStack.

### 2.5.3 Plateforme d'expérimentation LISP-Lab

Des travaux sur l'utilisation de LISP pour la gestion de réseaux virtuels et de datacenters sont conduits dans le cadre d'un projet français de recherche collaborative financé par l'ANR, le projet LISP-Lab **[19]**, piloté par le LIP6. Une plateforme d'expérimentation est ouverte à tout expérimentateur.

## Conclusion

Avec l'avènement des architectures « Spine and Leaf » et « BGP Top of Rack », les interconnexions de niveau 3 en Point-to-point se sont très largement développées ces dernières années, même en intra-datacenter. Mais il fallut néanmoins ajouter une couche d'abstraction afin de garder de vieilles habitudes de design et utilisations reposant sur le niveau 2.

Finalement, comme le disait déjà la RFC1925 en 1996 : « *It is easier to move a problem around (for example, by moving the problem to a different part of the overall network architecture) than it is to solve it* ». ■

## Remerciements

**Nous tenons à remercier nos collègues pour leurs précieuses relectures, ainsi que Jérôme Nicolle pour sa suggestion de plan.**

**Retrouvez toutes les références accompagnant cet article sur** http://www.miscmag.com/.