Gestion et interconnexion de fabrics eVPN/VxLAN

Oumar Niane, Alain Zamboni, Christophe Palanche, Fabrice Peraud

Gestion et interconnexion de fabrics IP EVPN/VXLAN

Oumar Niane
Direction du Numérique
14 rue René Descartes
67081 Strasbourg

Alain Zamboni
Direction du Numérique
14 rue René Descartes
67081 Strasbourg

Fabrice Peraud
Direction du Numérique
14 rue René Descartes
67081 Strasbourg

Christophe Palanché
Direction du Numérique
14 rue René Descartes
67081 Strasbourg

Résumé

Depuis la mise en exploitation du datacenter de l'université de Strasbourg (Unistra), la Direction du Numérique (DNum) a déployé plusieurs fabrics IP EVPN/VXLAN avec une architecture de type spine/leaf pour ses différents sites: datacenter, anciennes salles et PRA.

Pour permettre la propagation de services de niveau 2 ou 3 entre ces fabrics IP et vers d'autres futurs sites, nous avons dû considérer différentes approches d'interconnexion de datacenter (Data Center Interconnect – DCI) ayant chacune leurs avantages et leurs inconvénients. Dans l'article, nous présentons les trois scénarios de DCI que nous avons étudiés :

- *l'extension de la fabric IP d'un site à un autre;*
- *l'interconnexion par stitching de vlan (vlan handoff);*
- *l'utilisation d'EVPN gateway en bordure de la fabric IP.*

Les scénarios sont analysés d'après leur aspect technique. Nous expliquons nos choix dans différents cas d'usage et exposons, pour chacun, une analyse critique avec leurs avantages et leurs inconvénients.

Nous évoquons ensuite les modifications que nous avons apportées à nos outils pour gérer plusieurs fabric et pour garantir leur autonomie en termes d'exploitation.

Mots-clefs

réseau, datacenter, EVPN/VXLAN, fabric, DCI, PRA

1 Introduction

Lors de l'urbanisation réseau du *datacenter* de l'Université de Strasbourg (Unistra), la Direction du Numérique (DNum) a fait le choix de déployer une *fabric* IP EVPN[1]/VXLAN[2] avec une architecture de type *spine/leaf*. Cette même technologie a naturellement été choisie pour le site de PRA[3]. De même, des *fabrics* IP EVPN/VXLAN ont été déployées dans les anciennes salles informatiques afin de faciliter le déménagement des serveurs.

Pour permettre la propagation de services de niveau 2 ou 3 entre ces *fabrics* IP, nous avons dû considérer différentes approches d'interconnexion de *datacenter* (*Data Center Interconnect* - DCI) ayant chacune leurs avantages et leurs inconvénients.

Nous présenterons dans cet article les trois scénarios de DCI que nous avons été amenés à étudier :

- l’extension d’une *fabric* IP entre plusieurs sites ;
- l’interconnexion par *stitching* de vlan (VLAN *handoff*);
- l’utilisation d’EVPN *gateway* en bordure des *fabrics* IP.

Nous analyserons en détail ces différents scénarios d’un point de vue technique. Nous expliquerons les choix que nous avons faits pour les interconnexions des différentes *fabrics* IP en précisant leurs avantages et leurs inconvénients dans le contexte de l’Unistra.

Nous évoquerons ensuite les modifications que nous avons apportées à nos outils afin de pouvoir gérer plusieurs *fabrics*. Nous détaillerons les choix effectués pour garantir l’autonomie des *fabrics* en termes d’exploitation.

2 Rappel VXLAN / EVPN

VXLAN[1] est un réseau d’*overlay* (cf. Lexique) de niveau 2 sur un réseau de niveau 3. Il permet d’étendre des réseaux Ethernet sur une infrastructure réseau IP existante à travers des tunnels VXLAN dont les extrémités sont des interfaces appelées VTEP[4] (cf. Lexique).

EVPN[3] repose sur BGP[2] et ses extensions MP-BGP[4]. Utilisé comme plan de contrôle en association avec VXLAN, il permet la diffusion des informations MAC/VTEP et leur apprentissage.

Il existe plusieurs types de routes EVPN décrites en tableau 1.

1 Ethernet Virtual Private Network

2 Virtual eXtended LAN

3 Plan de Reprise d’Activité

4 VXLAN Tunnel EndPoint

Tableau 1: Types de routes EVPN

Route Type	Nom	Description
1	Ethernet Auto-Discovery	Permet la découverte automatique VNI[5]/VTEP/Segment Ethernet (cf. Lexique). Permet le "*mass withdraw*" (cf. Lexique) pour une convergence rapide
2	MAC/IP advertisement	Permet d'annoncer les MAC/IP pour les L2VPN
3	Inclusive Multicast	Permet d'annoncer l'existence d'un VNI sur une VTEP. Est utilisé pour peupler les tables de flood pour la gestion du BUM[6]
4	Ethernet Segment	Permet annoncer les VTEP appartement à un même segment Ethernet dans le cadre du multihoming (cf. Lexique). Est utilisé pour l'élection du DF[7] (cf. Lexique)
5	IP Prefix	Permet annoncer les préfixes IP pour les L3VPN
6 - 11		Les routes type 6 à 11 servent à la gestion du multicast

3 Modèles d'interconnexion

3.1 Extension de la *fabric* - Domaine EVPN unique

Dans ce modèle, il existe un domaine EVPN unique pour l'ensemble des sites. De même l'*underlay* et l'*overlay* sont uniques. Ainsi tous les services sont communs aux deux sites.

L'extension de l'*underlay* est effectuée par l'interconnexion des *border-leaves* de chaque site.

L'extension de l'*overlay*, quant à elle, est réalisée par l'établissement de session BGP EVPN entre les *spine* de tous les sites afin d'échanger leurs routes EVPN.

Il s'agit d'un seul domaine de gestion administrative imposant une gestion unique des VNI pour l'ensemble des sites.

Le domaine de réplication pour le trafic BUM est commun à tous les sites. Un VTEP dans un site verra l'ensemble des VTEP de tous les sites dans sa *flood-list*. Étant donné qu'EVPN utilise le mécanisme de réplication à la source du trafic BUM, le volume global de ce dernier peut s'avérer non négligeable.

5 Virtual Network Identifier

6 Broadcast Unknown Unicast Multicast

7 Designated Forwarder

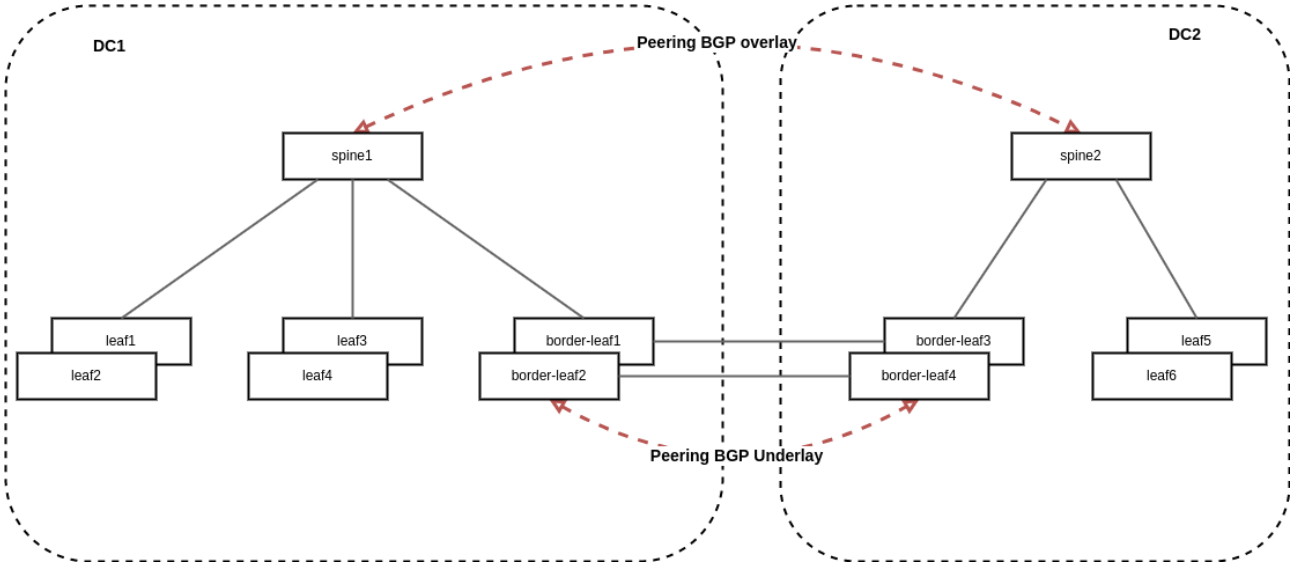

Figure 1: Fabric étendue

3.1.1 Avantages

Avec l'utilisation d'une *Fabric* étendue, nous avons pu recenser les avantages suivants :

- Simplicité de mise en œuvre et de configuration ;
- Encapsulation VXLAN de bout en bout ;
- Instanciation des services L2 et L3 de la même manière peu importe le site.

3.1.2 Inconvénients

L'inconvénient principal de ce scénario concerne les éléments de passage à l'échelle :

- nombre de routes EVPN ;
- nombre d'updates BGP ;
- gestion du BUM ;
- taille des *flood-lists*.

3.1.3 Utilisation à l'Unistra

Nous avons utilisé ce modèle d'interconnexion dans le cadre du déménagement des serveurs des anciennes salles vers le *datacenter*. Il s'est avéré particulièrement adapté de par sa simplicité de mise en œuvre et la facilité de gestion. Dans ce contexte, les services déployés dans les anciennes salles et le *datacenter* étaient identiques. De même, toutes les problématiques de passage à l'échelle ne se posaient pas, notre *fabric* se limitant à une centaine de *leaves*. De plus, l'interconnexion physique entre les salles se faisait via des fibres optiques nous appartenant.

3.2 Fabric DCI - Domaines EVPN multiples

Dans ce modèle, les deux domaines ont des plans de contrôle indépendants et les plans de données sont mis bout à bout via un *trunk* VLAN.

Les *border-leaves* marquent la limite du domaine EVPN de chaque site. Ces équipements sont connectés à une paire de *dci-leaves* via des *trunks* VLAN.

Le trafic entre les *border-leaves* et les *dci-leaves* n'est, par conséquent, pas encapsulé dans VXLAN.

Le domaine de réplication pour le trafic BUM est localisé dans chaque site. Un VTEP dans un site ne verra que les VTEP locaux du site dans sa liste *flood-list*. Cela réduit le volume global du trafic BUM traversant le DCI.

Pour la propagation d'un service L3 d'un site à un autre, l'interconnexion se fait grâce à un protocole de routage, par exemple BGP, au sein de la VRF.

De même, le domaine de gestion administrative est séparé entre les différents sites. L'association VLAN/VNI est interne à chaque *fabric*.

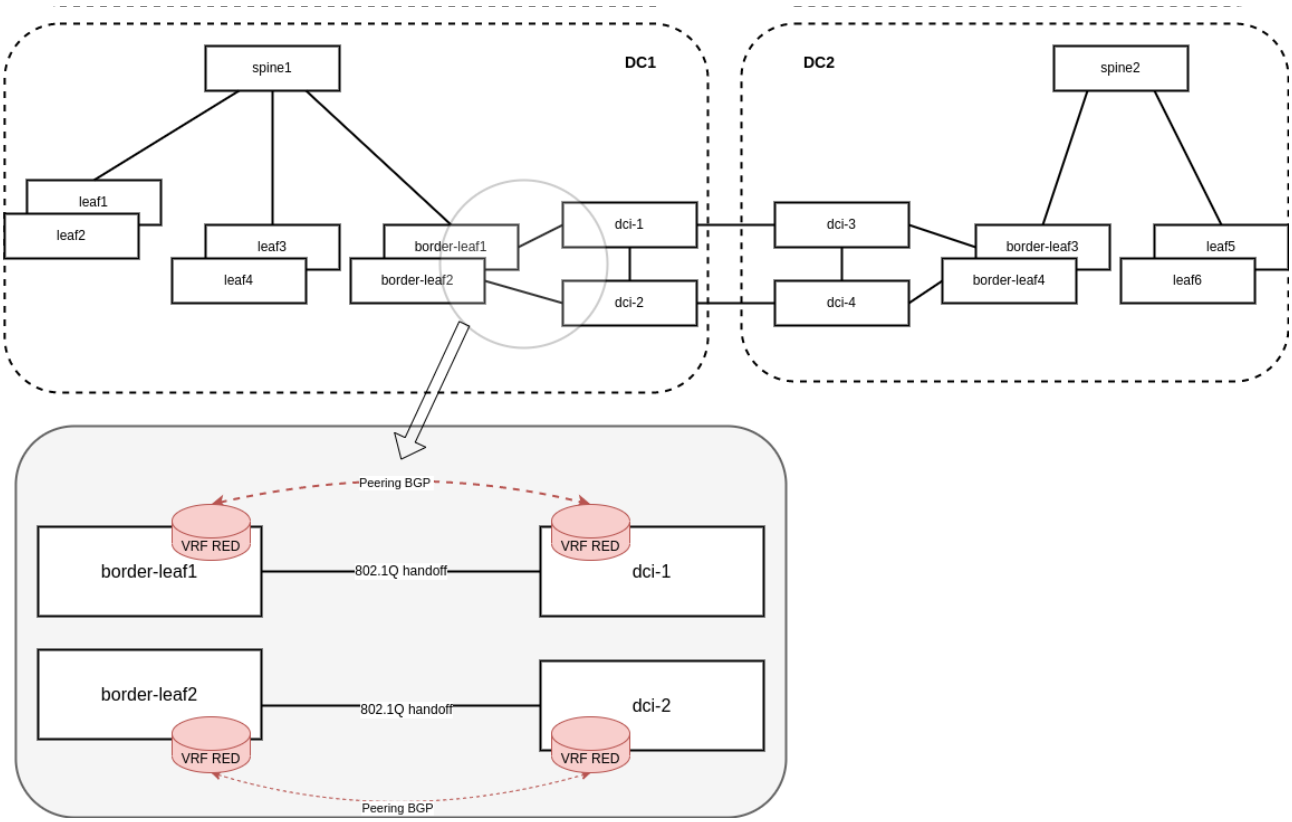

Figure 2: Fabric DCI intermédiaire

3.2.1 Avantages

La séparation en domaines EVPN distincts offre les avantages suivants :

- le domaine de réplication BUM est isolé dans chaque site ;
- la problématique du passage à l'échelle se pose par site et non globalement ;
- la flexibilité par rapport aux infrastructures existantes : interopérabilité facilitée par l'utilisation de protocoles bien répandus et largement supportés (*802.1q handoff*, *VRF handoff*) ;
- domaines de gestion administrative séparés.

3.2.2 Inconvénients

Ce scénario présente tout de même certains inconvénients :

- pas d’encapsulation VXLAN de bout en bout ;
- instanciation des services complexe notamment pour les L3 ;
- la nécessité de disposer d’équipements dédiés au DCI peut induire des coûts supplémentaires.

3.2.3 Utilisation à l’Unistra

Ce scénario est celui que nous avons retenu pour le déploiement et l’interconnexion du site de PRA de l’université situé à la Meinau.

Ce choix a principalement été guidé par souci d’isolation de nos sites. Du fait du faible nombre de services transportés entre le *datacenter* et le PRA, les inconvénients listés précédemment ne se font pas ressentir à l’heure actuelle.

Toutefois, dans le cadre du projet *datacenter* Grand Est, nous avons étudié le modèle d’interconnexion avec des EVPN *gateway*, qui simplifie notamment la configuration des services. Nous présentons cette solution ci-dessous.

3.3 Gateway EVPN

Dans ce modèle, les *border-leaves* jouent le rôle de *gateway* EVPN.

Les *gateways* des différents sites établissent des sessions BGP EVPN entre elles, pour annoncer leurs routes locales aux sites distants et les routes des sites distants en local. Dans les deux cas, elles se définissent comme *next-hop* lors de la ré-annonce. Ainsi, les *leaves* d’un domaine n’ont aucune connaissance des domaines distants.

Au niveau de VXLAN, les *gateways* disposent de deux *flood-list*, une pour le domaine local et une pour l’interconnexion des différents domaines contenant les VTEPs des *gateways* distants. Lorsqu'un *gateway* reçoit une trame VXLAN de son domaine local à destination d'un domaine distant et inversement, il effectue les opérations suivantes :

- désencapsulation de la trame Ethernet contenue dans VXLAN ;
- recherche du VTEP de destination dans ses tables locales ;
- ré-encapsulation dans VXLAN.

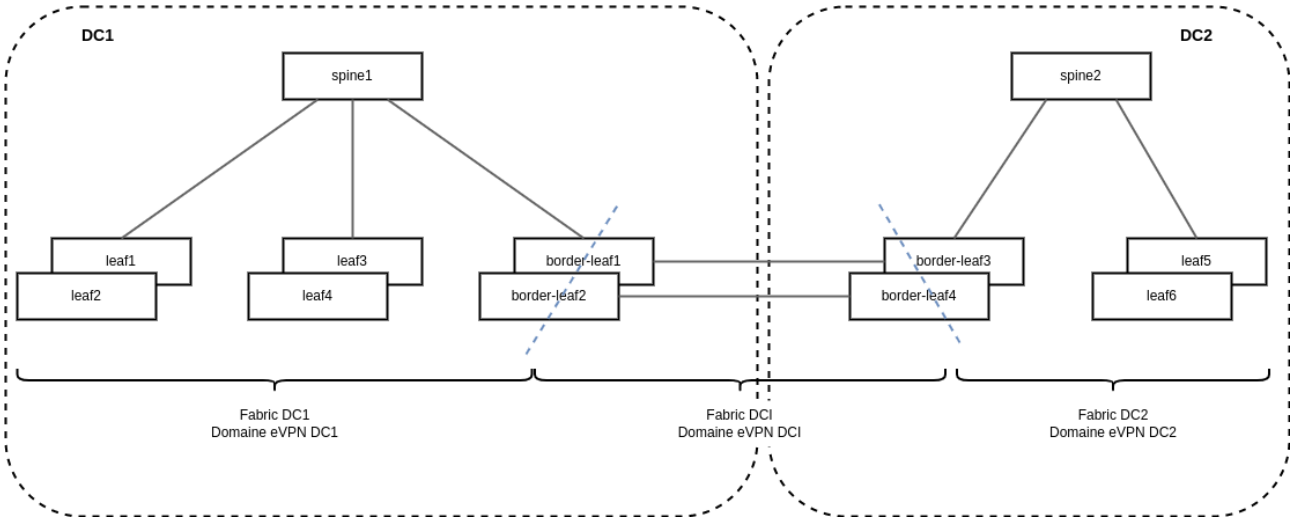

Figure 3: EVPN Gateway

3.3.1 Avantages

Grâce à son architecture hybride, ce scénario cumule plusieurs avantages des deux scénarios évoqués précédemment, à savoir :

- isolation de l'*underlay* : chaque site a son propre *underlay ;*
- hiérarchie de l'*overlay* : les EVPN *gateways* de chaque site font frontières (terminaison/origine des routes EVPN)
- domaine de flooding VXLAN séparé. Seules les *gateways* connaissent des VTEP des autres *gateways* en plus de leur domaine local ;
- la problématique du passage à l'échelle se pose par site et non globalement ;
- encapsulation VXLAN de bout en bout ;
- ne nécessite pas d'équipements et de *fabrics* dédiés au DCI.

3.3.2 Inconvénients

Nos manipulations avec cette technologie nous ont révélés les inconvénients suivants :

- configuration spécifique à la fonctionnalité de *gateway ;*
- problématique d'interopérabilité entre constructeurs dûe à des options dans la RFC ;
- pas encore de standard pour le prolongement des services L3, implémentation propriétaire par les constructeurs ;
- pas supporté sur tous les équipements : les *border-leaves* doivent supporter la rfc 9014.

3.3.3 Utilisation à l'Unistra

Ce scénario aurait été envisageable pour le déploiement de notre fabrique de PRA, mais cette technologie n'était pas encore supportée par nos équipements.

Il a depuis été évalué dans le cadre d'un POC avec l'Université de Lorraine pour l'interconnexion de nos deux *datacenter*. Dans ce contexte, nous avons tenté de mettre en œuvre la fonctionnalité *EVPN Gateway* entre deux *fabric* de maquette, une à Strasbourg composée d'équipements *Arista*, l'autre à Nancy avec des équipements *Juniper*.

Dans cette configuration, nous avons réussi à propager des services de niveau 2 entre les deux *fabric*. En revanche, au moment des tests, nous n'avons pas pu propager les services de niveau 3 avec *EVPN Gateway*, car les niveaux d'implémentation n'étaient pas les mêmes entre les deux constructeurs. Nous avons pu contourner cette limitation en utilisant IPVPN/MPLS[8] en complément.

Ensuite, pour tester la propagation de service de niveau 3 avec *EVPN Gateway*, nous avons déployé deux *border-leaves Arista* dans la *fabric* de Nancy. Cette configuration permettait également de tester l'interopérabilité des deux marques au sein d'une même *fabric*. Le bilan de cet essai est positif : l'utilisation de deux commutateurs *Arista* dans la *fabric Juniper* n'a pas posé de problème et la propagation des services de niveau 2 et 3 a fonctionné.

Ce scénario pourrait être une piste intéressante pour l'interconnexion des deux *datacenter*. Cependant, au moment de la rédaction du présent article, la jeunesse du standard et de ses implémentations n'en font pas un choix définitif. De plus, il nous reste à faire des tests pour valider certaines fonctionnalités, tel que l'*Anycast Gateway* (cf. Lexique).

8 https://datatracker.ietf.org/doc/html/rfc4364

4 Outils d'administration

Les outils d'administration utilisés pour l'exploitation de la *fabric datacenter* avaient été présentés dans notre contribution lors des JRES 2019[6]. Après un rappel des différents outils utilisés, nous évoquerons les différentes questions qui se sont posées dans la gestion de plusieurs *fabric*. Nous donnerons ensuite pour chaque outil la façon dont nous y avons répondu.

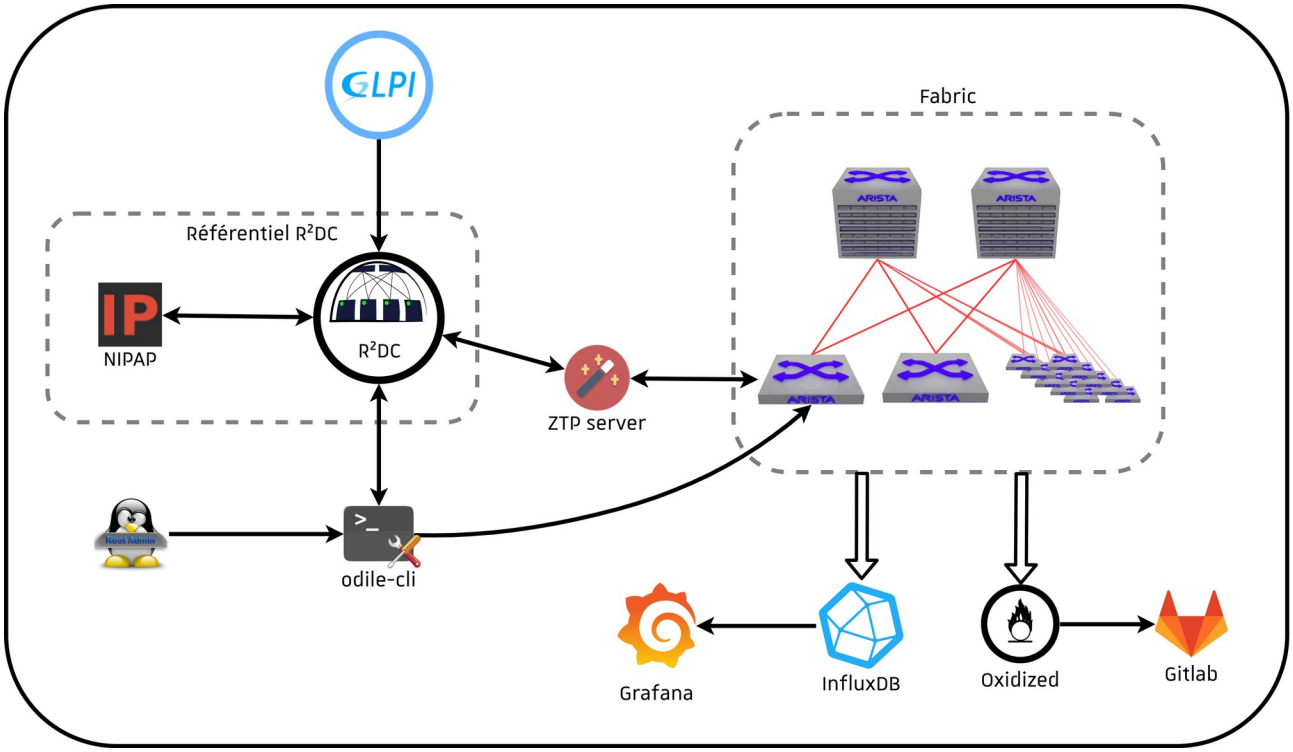

Figure 4: Outils d'exploitation

4.1 Description des outils

L'exploitation des *fabrics* de l'Unistra est assurée par un ensemble de logiciels issus de la communauté Open Source ou développés en interne. Ce schéma présente les différents outils mis en œuvre.

4.1.1 Référentiel R²DC

R²DC est le référentiel des *fabrics* de l'Unistra que nous avons développé. Il permet de gérer les éléments d'un réseau EVPN/VXLAN (équipements, services, préfixes, VRF, ASN, etc.). Il propose aujourd'hui une API REST développée en Python/Django reposant sur une base PostgreSQL pour le stockage des données. En plus des routes propres à ses objets, R²DC expose également des méthodes déclenchant des opérations dans l'IPAM NIPAP[7] (voir ci-dessous). L'API est décrite selon les normes OpenAPI 3 ce qui facilite la génération de librairies clientes REST dans différents langages.

4.1.2 NIPAP

NIPAP[7] est un IPAM open source. Dans notre architecture, il est utilisé comme backend de gestion des préfixes et VRF. Les clients n'accèdent pas directement à NIPAP, mais utilisent des routes exposées par R²DC. Ce choix a été effectué pour deux raisons :

- garantir la cohérence entre les informations stockées dans R²DC et NIPAP ;
- ajouter ultérieurement une gestion des droits sur les préfixes, fonction dont NIPAP ne dispose pas nativement.

4.1.3 Serveur ZTP

Le serveur permet aux équipements réseaux d'effectuer leur démarrage initial via le réseau hors-bande. Durant cette initialisation, les équipements sont mis à jour automatiquement et obtiennent une configuration, générée dynamiquement grâce aux informations stockées dans R²DC. Ce serveur ZTP est écrit en Python. Il est mis à disposition par Arista sur le dépôt github *arista-eosplus*[8] sous licence BSD 3.

4.1.4 Odile-cli

Odile-cli est un outil de configuration en ligne de commande des équipements de la *fabric* et de *R²DC*. Il permet ainsi de configurer les commutateurs tout en conservant la cohérence avec le référentiel. C'est un outil développé en interne en langage Go. Il communique directement avec les équipements et R²DC via leurs API respectives.

4.1.5 GLPI

GLPI[9] est utilisé pour inventorier les éléments d'infrastructures de la DNum, y compris les commutateurs des fabrics. R²DC l'utilise comme source de données pour ajouter les équipements à gérer et leurs caractéristiques (numéro de série, modèle).

4.1.6 Oxidized

Oxydized[10] est un logiciel de sauvegarde des configurations d'équipements réseaux. Il est distribué sous licence Apache v2.0.

4.1.7 Gitlab

À chaque changement de configuration détecté par *Oxidized*, ce dernier l'enregistre dans le dépôt git associé, les modifications sont ainsi stockées en in-band et consultables dans l'interface web de gitlab.

4.1.8 InfluxDB

Par l'intermédiaire d'un exécutable développé localement en langage *Go*, les commutateurs émettent des informations de télémétrie vers une base *InfluxDB*[11].

4.1.9 Grafana

Une instance *Grafana*[12] expose des tableaux de bords basés sur les données de la base *InfluxDB*. Cette instance est spécifique à la supervision du réseau et est accessible dans le réseau hors-bande.

4.2 État des lieux et problématiques

La première version des outils avait pour objectif de permettre l'exploitation de la *fabric* du *datacenter* principal de l'Unistra. La gestion multi-*fabric*, bien qu'identifiée comme nécessaire pour la gestion du site de PRA, avait été prévue ultérieurement pour respecter le planning de mise en service du site.

En parallèle, l'équipe projet Osiris 4 a opté pour le choix de la technologie EVPN/VXLAN pour le cœur de réseau Osiris. Ce choix a bien évidemment encouragé à la mutualisation des outils

d'administrations, ajoutant aux besoins non plus seulement la nécessité de gérer plusieurs *fabrics*, mais également deux périmètres d'administration.

Nous allons vous présenter ici quelques grandes lignes de nos réflexions. Ces éléments de réflexions sont interdépendants : une contrainte sur un des axes aura souvent une influence sur le choix à effectuer dans un autre.

4.2.1 Une technologie, mais deux périmètres d'administration différents

Bien que Osiris 4 et le réseau du *datacenter* partagent une même technologie, ils restent deux réseaux indépendants répondant à deux problématiques différentes. La mutualisation a donc dû permettre l'administration de réseaux par des équipes différentes avec des règles de gestion propres.

4.2.2 Cloisonnement et recouvrement de données

Certains des scénarios DCI évoqués précédemment rendent les *fabrics* suffisamment indépendantes pour que soit possible le recouvrement des valeurs de certains éléments. Par exemple, dans le cadre d'une architecture avec des domaines EVPN différents, les VNI de service sont propres à chaque *fabric*. Pour modéliser ce fonctionnement, le référentiel R²DC doit permettre l'instanciation de deux services différents avec le même VNI, chacun rattaché à sa *fabric* respective.

Pour chaque outil et données nous avions le choix entre :

- établir une règle de gestion empêchant le recouvrement (ex : création de plages réservées par *fabric*) ;
- déployer une nouvelle instance de l'outil;
- permettre le recouvrement de valeurs en cloisonnant par réseau/*fabric*.

4.2.3 Granularité d'instanciation des outils

Pour répondre aux besoins d'indépendance entre réseaux et *fabrics*, nous avons évalué pour chaque outil la pertinence ou nécessité de multiplier les instances d'un outil :

- une instance par *fabric* ;
- une instance pour les *fabrics* d'un même périmètre d'administration ;
- une instance commune à tous les périmètres d'administration.

4.3 Choix et actions pour chaque application

4.3.1 Référentiel R²DC/NIPAP

Pour gérer plusieurs *fabrics* avec R²DC, nous avons étudié différents scénarios.

Le premier consistait à créer une instance différente de R²DC/Nipap par *fabric*. Cette option n'a pas été retenue, car l'existence de plusieurs IPAM aurait alourdi la gestion. En effet, bien que les *fabrics* soient indépendantes sur plusieurs points, les plages d'IP de l'université restent communes et doivent être réparties entre les *fabrics*, sans recouvrement au sein d'une même VRF. Garder la cohérence des disponibilités de sous-réseaux entre plusieurs instances de NIPAP n'était pas envisageable.

Une alternative au premier scénario était d'avoir plusieurs instances de R²DC partageant une instance unique de NIPAP. Le lien fort entre Nipap et R²DC (les identifiants de préfixe NIPAP sont recensés dans l'objet R²DC correspondant), impose la cohérence constante des identifiants NIPAP entre toutes les instances de R²DC. Nous avons estimé que le rapport entre fonctionnement et complexité d'exploitation ne serait pas intéressant.

La solution qui nous a paru la plus optimale consistait à modifier le modèle de données de R²DC pour gérer la notion de *fabric*. Nous avons donc modélisé un objet *fabric* auquel nous avons lié tous les objets l'exigeant : équipements, services, ESI, etc.

4.3.2 Serveur ZTP

Nous avons mutualisé le serveur ZTP pour l'ensemble des *fabrics*, car il sait gérer nativement des *template* de configuration et qu'il n'est utilisé que lors du déploiement initial des équipements. Nous avons adapté le plug-in R²DC pour veiller à ce que les éléments de configuration soient liés à la *fabric* du commutateur s'initialisant.

4.3.3 Odile-cli

Les modifications opérées sur le modèle du référentiel R²DC ont amené à revoir notre outil de configuration des commutateurs. Nous avons ainsi ajouté une nouvelle option permettant de spécifier la *fabric* sur laquelle nous travaillons pour chaque exécution de commande. Dans le fichier de configuration, une nouvelle syntaxe permet maintenant de définir, pour chaque *fabric*, les informations d'authentification aux équipements et de déclarer une *fabric* par défaut.

4.3.4 Oxidized

Historiquement, le réseau Osiris bénéficiait déjà d'une instance d'Oxidized qui dépassait le cadre du backbone et sauvegardait également les configurations des commutateurs d'extrémités. Afin de conserver une séparation administrative claire entre l'exploitation des deux réseaux, nous avons décidé de créer une deuxième instance dédiée au réseau du *datacenter*.

4.3.5 Gitlab

L'instance Gitlab est celle commune à l'Unistra. Nous avons créé un dépôt respectif pour chaque instance d'Oxidized, rangé dans un namespace lui aussi dédié au réseau correspondant, ceci pour pouvoir gérer des droits d'accès assez finement.

4.3.6 InfluxDB/Grafana

Nous avons jugé préférable d'avoir une instance d'InfluxDB différente pour le *datacenter* et Osiris afin de conserver un paramétrage spécifique à chaque périmètre d'administration. L'utilisation de Grafana nous a confortés dans notre choix, celui-ci permettant d'interroger plusieurs bases de données et de croiser leurs informations dans les tableaux de bord.

4.3.7 GLPI

GLPI est utilisé pour inventorier le matériel d'infrastructure. De ce fait, il est intéressant d'y faire figurer l'appartenance d'un équipement à un réseau, mais sans aller au détail de la *fabric*. Nous avons donc utilisé le champ *type* de l'objet "Équipement réseau" pour effectuer la répartition entre réseau Osiris 4 ou le réseau du DC. En fonction de cette information les équipements sont importés dans *R²DC* soit dans la *fabric O4* pour les équipements Osiris, soit dans la *fabric DC* pour les équipements *datacenter*. C'est au moment de la préparation de l'équipement qu'il sera placé dans la bonne *fabric* par l'opérateur.

5 Conclusion

Il existe plusieurs scénarios pour interconnecter des *fabrics*. Nous en avons testé et présenté trois. Chacun, par ses avantages et ses inconvénients, s'adapte plus ou moins à votre contexte.

À l’Unistra, nous avons dû prendre en considération des paramètres comme le temps de mise en œuvre, la complexité, le nombre de services à étendre et de sites à interconnecter.

La technologie est à l’heure actuelle en cours d’évolution. De nouvelles normes apparaissent régulièrement (ex : RFC9014[13] – mai 2021) et étendent les possibilités d’interconnexions. Malheureusement, celles-ci ne sont pas implémentées au même rythme par tous les constructeurs, ce qui peut poser des problèmes d'interopérabilité.

L’interconnexion de *fabric* ne se limite pas à une problématique technologique. Les outils d’exploitation doivent être pensés ou adaptés pour une gestion sur des périmètres distincts ou par des équipes différentes.

Nous prévoyons aussi d'étudier d'autres évolutions techniques ou logicielles.

Comme évoqué précédemment, des travaux sont en cours avec l’université de Lorraine pour l’interconnexion de nos *datacenter* respectifs dans le cadre d’un projet régional. Actuellement, nous affinons conjointement des scénarios pour permettre à chaque site de devenir le PRA de l’autre et simplifier l’extension des services entre les sites.

D’autre part, le futur changement de l’IPAM de l’Unistra va nous amener à remettre en question l’utilisation de NIPAP au sein de R²DC. Enfin, pour améliorer le service aux usagers du *datacenter*, nous souhaitons exposer les outils d’administration des *fabrics* à travers une interface web qui offrira à chaque client une gestion autonome des ports et des services sur son périmètre.

Lexique

AS : Autonomous System. Ensemble de réseaux IP sous le contrôle d’une seule et même entité.

ASN : Autonomous System Number. Identifiant d’un AS. Encodé sur 16 bits ou 32 bits.

Osiris : le réseau métropolitain strasbourgeois de l'enseignement supérieur et de la recherche.

Underlay : réseau physique sur lequel s’appuie le réseau d’.

Overlay : réseau virtuel transporté sur des liens logiques du réseau sous-jacent (*underlay*).

VNI : Virtual Network Identifier. Identifiant de réseau virtuel VXLAN dans le plan de données. Encodé sur 24 bits.

Mass-withdraw : mécanisme permettant de notifier rapidement à l'ensemble des VTEP l'indisponibilité d'un Segment Ethernet pour qu'ils invalident les routes associées à ce dernier.

Multihoming : connexion d’un client à plusieurs VTEP en actif/passif ou actif/actif.

ESI : Ethernet Segment Identifier. Identifiant pour un groupe de liens appartenant au même segment Ethernet.

IPAM : IP Address Management. Méthodologie de gestion des adresses IP.

VTEP : VXLAN Tunnel End Point. Interface de terminaison d’un tunnel VXLAN

Flood-list : Liste des VTEP auxquelles le trafic BUM doit être envoyé pour un service donné

DF : Designated Forwarder. VTEP propageant le trafic BUM vers l’extérieur dans le cadre d’une architecture de type multi-homing (double attachement de niveau 2) active/active.

Anycast Gateway : Fonctionnalité permettant de distribuer la passerelle d’un réseau sur plusieurs VTEP, permettant ainsi un routage au plus proche.

REST : Representational State Transfer. Style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web.

Bibliographie

[1] RFC7348 – Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks
https://datatracker.ietf.org/doc/rfc7348

[2] RFC4271 – A Border Gateway Protocol 4 (BGP-4)
https://datatracker.ietf.org/doc/rfc4271

[3] RFC8365 – A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN) ;
https://datatracker.ietf.org/doc/rfc8365

[4] RFC 4760 – Multiprotocol Extensions for BGP-4
https://datatracker.ietf.org/doc/html/rfc4760

[5] RFC7432 – BGP MPLS-Based Ethernet VPN
https://datatracker.ietf.org/doc/rfc7432

[6] Christophe Palanché, Alain Zamboni, Fabrice Peraud, Oumar Niane
Le réseau Datacenter Unistra
Dans Actes du congrès JRES 2019, Dijon, décembre 2019
https://conf-ng.jres.org/2019/document_revision_5662.html?download

[7] NIPAP
https://spritelink.github.io/NIPAP/

[8] EOSPLUS/ZTPSERVER
https://github.com/arista-eosplus/ztpserver

[9] GLPI ITSM Gestion de Services Informatiques - Open Source
https://glpi-project.org/fr/

[10] Oxidized
https://github.com/ytti/oxidized

[11] InfluxDB
https://www.influxdata.com/products/influxdb/

[12] Grafana: The open observability platform
https://grafana.com/

[13] RFC9014 - Interconnect Solution for Ethernet VPN (EVPN) Overlay Networks
https://datatracker.ietf.org/doc/rfc9014