RFC 7665 : Service Function Chaining (SFC) Architecture

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 octobre 2015

Date de publication du RFC : Octobre 2015

https://www.bortzmeyer.org/7665.html

Aujourd'hui, des flots de paquets IP qui voyagent d'un point à un autre de l'Internet ont peu de chances de le faire sans passer par divers traitements, des plus honorables (statistiques, lutte contre une attaque par déni de service) aux moins avouables (étranglement du trafic pour pousser un fournisseur de contenu à payer, censure). Actuellement, ces traitements sont rigidement liés à la topologie du réseau sous-jacent et manquent donc de souplesse. Dès qu'on veut chaîner plusieurs traitements, le résultat de la configuration devient difficile à lire et à maintenir (le problème est décrit plus en détail dans le RFC 7498 [1]). L'idée de base du projet SFC ("*Service Function Chaining*") est de spécifier une architecture (ce RFC) et peut-être plus tard des protocoles pour organiser ces traitements de manière plus souple.

Le RFC 7498 avait très bien décrit le problème qu'on essaye de résoudre. Ce nouveau RFC 7665 est plus concret, décrivant l'architecture des SFC mais je trouve personnellement qu'il reste encore très abstrait et qu'il ne fait pas beaucoup avancer la compréhension, par rapport à son prédécesseur.

Donc, l'architecture des SFC ("*Service Function Chaining*"). On veut avoir des chaînes ordonnées des traitements appliqués aux flux réseau, et on veut pouvoir les créer, les modifier et les détruire facilement, avec bien plus de souplesse que le système actuel (section 1 du RFC).

Les paquets entrent et sont **classés**, les traitements étant ensuite appliqués en fonction du classement. Ces traitements sont appliqués dans l'ordre de la chaîne et, attention, il peut y avoir une re-classification dans un des ces traitements, menant à une autre chaîne.

Pour définir l'architecture, on suppose que :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7498.txt

— Les traitements sont tellement variés qu'il est impossible de les normaliser ou même de les décrire de manière exhaustive. On va donc, dans le projet SFC, s'intéresser uniquement à l'ordonnancement de ces traitements, pas au traitement lui-même. Chaque traitement est une « boîte noire ».
— La liste des traitements appliqués, les critères d'application, et l'ordre d'application, sont des décisions purement locales. Il n'y aura pas de normalisation des traitements (du genre « il faut commencer par un filtrage IP »).
— L'application de ces traitements en chaîne dépend évidemment du réseau sous-jacent qui va déplacer les bits d'une machine à l'autre mais ce réseau n'est pas décrit par SFC, chacun peut utiliser ce qu'il veut.

Chaque traitement (on le nomme un SF pour "*Service Function*") est identifié de manière unique dans le domaine (dans les exemples du RFC, l'identificateur est un simple nombre). Une suite de SF est une SFC ("*Service Function Chain*"). Une SFC peut être unidirectionnelle (par exemple des paquets qui sont d'abord comptés, puis filtrés, puis modifiés) ou bidirectionnelle.

Un SFF ("*Service Function Forwarder*") est responsable d'acheminer le trafic d'une SF à une autre. Un SFP ("*Service Function Path*") est la réalisation concrète d'une SFC et un RSP ("*Rendered Service Path*") est la suite des SF effectivement visités. Prenons comme exemple une SFC qui dit que le trafic doit passer par deux SF, un pare-feu et un cache. Le SFP, plus concret, va indiquer quelle instance des deux DF, et dans quel centre de données, on va utiliser. Et le RSP indiquera où le trafic est réellement passé (il peut différer du SFP si ce dernier laissait une certaine latitude, ou bien si quelque chose ne s'est pas passé comme prévu).

La section 3 du RFC liste les grands principes, notamment :
— Indépendance par rapport à la toplogie du réseau : pas besoin de recâbler quand on modifie une SFC.
— Séparation entre les SFP et la transmission des paquets par les routeurs.
— Classification des paquets, qui décidera du SFP où on enverra le trafic.
— Métadonnées accessibles, par exemple le résultat de la classification sera visible par les SF (voir plus loin la notion d'encapsulation).
— Indépendance des SFC entre elles, et de la SFC par rapport aux SF, nécessaire à la souplesse du dispositif.

La section 4 de notre RFC détaille les composants du système vus plus haut (SF, SFF, etc). Ces composants sont reliés par l'encapsulation SFC, qui définit comment on va encapsuler le trafic pour l'acheminer d'un composant à l'autre. Attention, ce n'est pas un remplaçant d'IP : lorsque les SF sont sur des machines séparées, il faudra toujours un réseau sous-jacent pour transmettre les paquets. Le plan de contrôle (comment trouver où acheminer le trafic) n'est pas décrit, il est hors-sujet pour ce projet, qui reposera sur des protocoles déjà existants.

D'ailleurs, en parlant de réseau, la section 5, qui couvre divers problèmes de réalisation, note entre autres que qui dit encapsulation dit problèmes de MTU, suite aux octets d'en-tête de l'encapsulation. Il n'y a pas de solution idéale pour les futurs protocoles concrets qui instancieront cette vision abstraite des "*Service Function Chains*" mais il faut au moins garder ce problème en tête.

Autre problème pratique dans la même section 5, la fiabilité. Si on met des services supplémentaires dans le réseau, cela ne doit pas trop diminuer sa résilience. Par exemple, prenons une SFC qui ne comporte qu'un seul SF, chargé de mesurer le trafic. La panne de l'équipement qui porte ce SF ne doit pas arrêter tout le trafic. Ici, c'est relativement facile car le SF « compteur » n'a pas besoin d'être en série sur le trajet, il peut être placé en dérivation. Mais d'autres SF ont besoin de modifier le trafic et donc doivent être placés en série. Il faut alors prévoir des solutions de secours, par exemple à base de VRRP (RFC 5798).

Le déploiement des SFC peut aussi avoir des conséquences sur la sécurité. On passe d'un réseau assez statique avec peu de fonctions à un réseau très dynamique pouvant assurer plein de fonctions (section 6 de notre RFC).