

# RFC 3654 : Requirements for Separation of IP Control and Forwarding

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 18 juillet 2007

Date de publication du RFC : Novembre 2003

<https://www.bortzmeyer.org/3654.html>

---

Ce RFC marque le début d'un processus très ambitieux, qui doit permettre l'arrivée de la normalisation non plus seulement entre les nœuds de l'Internet mais aussi à l'intérieur de ces nœuds. Le groupe de travail IETF ForCES <<http://www.ietf.org/html.charters/forces-charter.html>> cherche en effet à normaliser la communication entre les différents composants d'un routeur IP. Ce premier RFC du groupe est le cahier des charges du projet.

Un routeur est composé notamment d'un "*forwarding engine*" qui effectue le traitement de chaque paquet (décider où l'envoyer, puis le faire) et d'un "*control engine*" ou "*routing engine*" qui traite les protocoles de routage comme BGP ou OSPF, les statistiques, les réponses aux requêtes SNMP, etc. Sur un routeur de haut de gamme, par exemple sur les Juniper ou sur certains Cisco, la séparation entre ces deux moteurs est très avancée : le "*forwarding engine*" est mis en œuvre par des circuits matériels spécialisés, les ASIC, alors que le "*routing engine*" est un logiciel bien plus traditionnel, tournant typiquement sur un processeur ordinaire (un Pentium sur les Juniper, par exemple, en dépit de leur prix). Si on utilise un ordinateur ordinaire comme routeur, les deux moteurs sont tous les deux mis en œuvre en logiciel, dans le processeur, mais sont néanmoins séparés dans le code. Par exemple, sur une machine Unix, le "*forwarding engine*" est le noyau Unix et le "*routing engine*" est un programme en mode utilisateur, comme Quagga ou Xorp.

Les deux moteurs ont des demandes très distinctes ; besoin de RAM et d'un vrai processeur pour le "*control engine*", qui doit faire des calculs complexes ; capacité à router à la "*line speed*" donc plutôt temps réel, pour le "*forwarding engine*".

La communication entre ces deux moteurs se fait à chaque fois avec un protocole privé, souvent non documenté lorsqu'il s'agit de logiciel non-libre. Sur Linux, le protocole se nomme Netlink et est

documenté dans le RFC 3549<sup>1</sup>. Sur FreeBSD, cela se fait avec les *"routing sockets"* documentées dans `route(4)`. Du fait du caractère privé de ces protocoles, un logiciel comme Quagga doit gérer de nombreux cas différents pour communiquer avec le noyau et une carte d'interface d'un Cisco ne peut pas être mis dans un Juniper : même si elles étaient matériellement compatibles, la carte ne pourrait pas communiquer avec le *"routing engine"*, faute d'un protocole commun.

D'où l'idée d'un protocole normalisé de façon à ce que, comme le dit le RFC, *"A standard set of mechanisms for connecting these components provides increased scalability and allows the control and forwarding planes to evolve independently, thus promoting faster innovation."* Notre RFC est le cahier des charges de ce futur système.

Que dit-il ? Il définit dans sa section 2 un vocabulaire standard, avec les notions de *"Network Element (NE)"* (une machine complète, vue de l'extérieur comme une entité unique), *"Forwarding Element (FE)"* (un dispositif qui sert à faire suivre les paquets) et *"Control Element (CE)"* (un dispositif qui sert à gérer les protocoles de routage). Le système ForCES concernera la communication entre FE et CE, communication qui ne passera pas forcément par IP.

ForCES conviendrait aussi bien à des cas où les éléments sont dans une même boîte (cas des routeurs actuels), avec transport des messages ForCES sur le bus interne, qu'au cas où les éléments sont physiquement séparés, avec transport des messages ForCES sur TCP/IP et Ethernet.

Ensuite, la section 5 explique la nécessité de développer un **modèle** pour les entités gérées par ForCES. En effet, il existe de nombreuses sortes de routeurs, ayant des capacités très différentes. Pour que les FE et CE puissent se parler, il faut qu'il existe un modèle de leurs capacités, pour qu'ils puissent savoir quoi attendre de l'autre.

Enfin la section 6 décrit les exigences pour le protocole lui-même.

ForCES permettrait de résoudre le problème des routeurs Unix actuels : ils tournent typiquement sur PC, une machine qui convient parfaitement comme CE mais qui est faible en FE. La partie la plus politiquement sensible, le CE, pourrait être en logiciel libre et sous-traiter le routage effectif à des ASIC ou à des boîtiers spécialisés, situés hors du PC.

À l'heure d'aujourd'hui, bien que très avancés, les textes qui normaliseront le modèle et le protocole ForCES sont toujours à l'état d'*"Internet-Drafts"*. Le travail s'est révélé très, peut-être trop ambitieux. Seul est sorti le RFC 3746, qui n'est encore qu'une description assez générale de l'architecture.

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc3549.txt>