

RFC 7557 : Extension Mechanism for the Babel Routing Protocol

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 juin 2015

Date de publication du RFC : Mai 2015

<https://www.bortzmeyer.org/7557.html>

Le protocole de routage Babel, spécifié dans le RFC 6126¹, est désormais doté d'un mécanisme d'extension documenté. Il a depuis été intégré dans la norme Babel, dans le RFC 8966.

Un paquet Babel comprend un en-tête fixe, suivi d'une série de TLV. Babel prévoyait des mécanismes d'extension en réservant certaines valeurs et en précisant le comportement d'un routeur Babel lors de la réception de valeurs inconnues. Ainsi :

- Un paquet Babel avec un numéro de version différent de 2 doit être ignoré, ce qui permet de déployer une nouvelle future version de Babel sans que ses paquets ne cassent les implémentations existantes,
- Un TLV de type inconnu doit être ignoré (RFC 6126, section 4.3), ce qui permet d'introduire de nouveaux types de TLV en étant sûr qu'ils ne vont pas perturber les anciens routeurs,
- Les données contenues dans un TLV au-delà de sa longueur indiquée, ou bien les données présentes après le dernier TLV, doivent également être silencieusement ignorées (au lieu de déclencher une erreur). Ainsi, une autre voie d'extension est possible, où on pourra glisser des données supplémentaires.

Bien sûr, ces règles ne garantissent pas à elles seules que les extensions cohabiteront bien. D'où ce nouveau RFC, qui décrit comment utiliser ces possibilités, sans conflit entre deux extensions différentes.

La section 2 décrit toutes les possibilités d'extensions propres. Cela commence par une nouvelle version du protocole (l'actuelle version est la 2), qui utiliserait des numéros 3, puis 4...

Moins radicale, une extension de la version 2 peut introduire de nouveaux TLV (qui seront ignorés par les mises en œuvre anciennes de la version 2). Ces nouveaux TLV doivent suivre le format de la section 4.3 du RFC 6126.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc6126.txt>

Si on veut ajouter des données dans un TLV existant, en s'assurant qu'il restera correctement analysé par les anciennes mises en œuvre, il faut jouer sur la différence entre la taille explicite ("*explicit length*") et la taille effective ("*natural length*") du TLV. La taille explicite est celle qui est indiquée dans le champ `Length` spécifié dans la section 4.3 du RFC 6126. La taille effective est celle déduite d'une analyse des données (plus ou moins compliquée selon le type de TLV). Comme les implémentations de Babel doivent ignorer les données situées après la taille naturelle, on peut s'en servir pour ajouter des données. Elles doivent être encodées sous forme de sous-TLV, chacun ayant type, longueur et valeur (leur format exact est décrit en section 3).

Babel a aussi six bits libres dans les TLV de type `Update` (champ `Flags`, RFC 6126, section 4.4.9). Notre RFC déconseille de les utiliser, pour l'instant.

Enfin, après le dernier TLV (Babel est transporté sur UDP, qui indique une longueur explicite), on peut encore ajouter des données. L'idée était d'y mettre une signature cryptographique mais le système d'authentification de Babel, spécifié dans le RFC 8967, se sert plutôt de nouveaux TLV. Cet espace potentiel est donc inutilisé pour l'instant.

Bon, mais alors quel mécanisme d'extension choisir ? La section 4 fournit des pistes aux développeurs. Le choix de faire une nouvelle version est un choix drastique. Il ne devrait être fait que si la nouvelle version est réellement incompatible avec la précédente.

Un nouveau TLV, ou bien un nouveau sous-TLV d'un TLV existant est la solution à la plupart des problèmes d'extension. Par exemple, si on veut mettre de l'information supplémentaire aux mises à jour de routes (TLV `Update`), on peut créer un nouveau TLV « `Update` enrichi » ou bien un sous-TLV de `Update` qui contiendra l'information supplémentaire. Attention, les conséquences de l'un ou l'autre choix ne seront pas les mêmes. Un TLV « `Update` enrichi » serait totalement ignoré par un Babel ancien, alors qu'un TLV `Update` avec un sous-TLV d'« enrichissement » verrait la mise à jour des routes acceptée, seules l'information supplémentaire serait perdue.

Il existe désormais, pour permettre le développement d'extensions, un registre IANA des types de TLV <<https://www.iana.org/assignments/babel/babel.xhtml#tlv-types>> et un des sous-TLV <<https://www.iana.org/assignments/babel/babel.xhtml#sub-tlv-types>> (section 5 du RFC) et plusieurs extensions s'en servent déjà.