

RFC 5225 : RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP Lite

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 mai 2008

Date de publication du RFC : Avril 2008

<https://www.bortzmeyer.org/5225.html>

ROHC est un mécanisme de description des compressions possibles des en-têtes des paquets IP. Ce RFC décrit les profils de compression qui peuvent être utilisés pour un certain nombre des protocoles les plus courants (la définition de ROHC lui-même étant dans le RFC 5795¹, la section 3 de notre RFC fait quelques rappels).

Mettant à jour les RFC précédents, le RFC 3095, mais aussi les RFC 3843 et RFC 4019, ce RFC est l'état de l'art en ce qui concerne la compression d'en-têtes. La principale innovation est l'utilisation du langage formel ROHC-FN du RFC 4997 (la section 4.2 résume les principaux changements).

Le gros du RFC commence avec la section 5 qui expose les principes de chaque profil. Un canal ROHC relie un **compresseur** et un **décompresseur**. Le compresseur indique les changements par rapport aux prévisions, l'absence de changements signifiant au décompresseur qu'il peut appliquer la règle par défaut. Les systèmes de compression en réseau doivent toujours faire un compromis entre la robustesse (qui nécessite de la redondance, donc de ne pas trop comprimer) et l'efficacité (qui impose de supprimer toute la redondance). Voir par exemple la section 5.1.2, qui discute ce compromis.

Enfin, la section 6 contient les profils eux-mêmes, exprimés en partie en ROHC-FN. Cette section contient des **fonctions** de compression et les règles ROHC-FN qui les utilisent. Parmi les fonctions primitives, qui ne sont pas en ROHC-FN, la section 6.6.6 (fonction `inferred_ip_v4_length`) dit comment comprimer la champ "*Length*" d'IPv4. Étant déductible de la longueur du datagramme, il n'est pas indispensable et peut donc être comprimé à une taille de... zéro bits. La section 6.6.9 décrit une méthode bien plus complexe (fonction `timer_based_lsb`) pour les champs dont la valeur change approximativement en fonction linéaire du temps (elle nécessite donc que les horloges du compresseur et du décompresseur ne dérivent pas trop l'une par rapport à l'autre).

Un exemple en ROHC-FN est au contraire la définition de la fonction `scaled_ts_lsb` (qui sert pour RTP), qui s'exprime ainsi :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5795.txt>

```
scaled_ts_lsb(time_stride_value, k)
{
  UNCOMPRESSED {
    timestamp [ 32 ];
  }

  COMPRESSED timerbased {
    ENFORCE(time_stride_value != 0);
    timestamp := timer_based_lsb(time_stride_value, k,
                                  ((2^k) / 2) - 1);
  }

  COMPRESSED regular {
    ENFORCE(time_stride_value == 0);
    timestamp := lsb(k, ((2^k) / 4) - 1);
  }
}
```

L'annexe A est une excellente étude des en-têtes de divers protocoles, avec une classification de leur comportement, selon qu'il permet la compression ou pas. C'est une lecture recommandée pour tous ceux qui s'intéressent à la compression d'en-têtes mais n'ont pas envie de relire le RFC original du protocole. Ils trouveront résumées de façon claire les caractéristiques de chaque champ de l'en-tête. Par exemple, pour IPv6 (annexe A.2) :

- *"Version"* is *STATIC-KNOWN* c'est-à-dire constante pour toute la durée du flot et à une valeur « bien connue », en l'occurrence 6.
- *"Hop Limit"* est *RACH* (*"RArely CHanging"*), c'est-à-dire changeant rarement (ce nombre ne bouge que lorsque les routes sont modifiées)
- *"Source Address"* est *STATIC-DEF*, c'est-à-dire constant pour toute la durée du flot et pouvant d'ailleurs servir à définir le flot.

Il n'existe apparemment pas d'implémentation libre de ROHC. Mais plein de déploiements sont déjà faits dans les téléphones portables. La thèse d'Ana Minaburo <<http://www.rennes.enst-bretagne.fr/~minaburo/these.pdf>> contient une bonne introduction à ROHC.