RFC 5537 : Netnews Architecture and Protocols

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 29 novembre 2009

Date de publication du RFC : Novembre 2009

http://www.bortzmeyer.org/5537.html

L'antique RFC 1036[1] qui, pendant les vingt dernières années, était la seule norme gouvernant le système de "*news*", vient d'être remplacé par une série de RFC dont celui qui fait l'objet de cet article, le RFC 5537 qui décrit l'architecture générale des "*news*".

Les "*news*" (ou "*Netnews*", dit le RFC) sont un extraordinaire système de distribution de messages, organisés en **groupes de discussion** ("*newsgroups*"). Chacun de ces groupes peut être un vecteur d'annonces ou bien un forum de discussion, où la vigueur de ces discussions sont une caractéristique fréquente des "*news*". Les "*news*" existaient avant que l'Internet ne se répande et, jusqu'à très récemment, était souvent distribuées avec des protocoles non-TCP/IP tel qu'UUCP. Mille fois, la fin des "*news*" a été annoncée, au profit de gadgets récents accessibles via le Web et à chaque fois, elles ont continué à inonder les serveurs et à passionner des dizaines de milliers de participants. À l'époque où n'importe quel système de communication via le Web fait immédaitement l'objet de l'attention des médias et des experts, les "*news*" reste un outil « invisible » (en tout cas invisible aux chefs, aux experts, aux ministres et aux journalistes), ce qui fait une bonne partie de leur charme.

Maintenant, place à la technique. Comment les "*news*" fonctionnent-elles (section 1 de notre RFC) ? Les serveurs de "*news*" s'échangent des articles, dans un format normalisé. Les articles sont regroupés en groupes comme `fr.reseaux.internet.hebergement` ou `soc.culture.belgium`. Les groupes qui partagent un préfixe commun comme `fr` forment une hiérarchie. Chaque serveur transmet ses articles à ses voisins qui, à leur tour, le transmettent à leurs voisins, un protocole dit d'inondation (il existe d'autres protocoles fondés sur le bavardage de proche en proche <http://www.bortzmeyer.org/gossip-protocol.html> dans l'Internet). Un ensemble de serveurs ainsi reliés est un réseau et le principal se nomme Usenet (il existe aussi des réseaux privés, bien plus petits). Lorsque les gens disent qu'ils lisent les "*news*" ils font en général allusion à celles d'Usenet.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc1036.txt`

Les deux principaux RFC de la nouvelle série, celle qui remplace le RFC 1036, sont le RFC 5536, qui normalise le format des messages, un format très inspiré de celui du courrier électronique, et notre RFC 5537 qui décrit le cadre général. L'annexe A est consacrée aux changements depuis le RFC 1036. Le développement de cette nouvelle série de RFC a finalement pris près de dix ans. (Une documentation du premier effort se trouve dans le RFC 1849, « "*Son of RFC 1036*" ».)

Parmi les serveurs de "*news*", la section 1.2 fait la distinction entre :
— "*injecting agent*" le premier serveur qui reçoit un article, article à qui il manquera peut-être quelques caractéristiques pour être tout à fait conforme à la norme,
— "*relaying agent*" un serveur qui recevra et transmettra un article, typiquement sans le modifier,
— "*serving agent*", un serveur qui permettra la lecture par les clients finaux,
— "*user agent*", le logiciel qui tourne sur le bureau de l'utilisateur final, lui-même séparé en "*posting agent*" qui permet d'écrire et "*reading agent*" qui permet de lire,
— et les passerelles, qui connectent le monde des "*news*" à d'autres systèmes, comme le courrier électronique ou le Web
mais, en pratique, beaucoup de logiciels assurent plusieurs fonctions à la fois. Par exemple, INN est "*injecting agent*", "*relaying agent*" et "*serving agent*".

Les devoirs des différents logiciels sont résumés dans la section 3. Il y a des grands principes (section 3.1), comme le Principe de Robustesse « Soyez restrictif dans ce que vous envoyez et ouvert pour ce que vous recevez » ou bien comme le principe d'Hippocrate, « Avant tout, ne pas nuire », c'est-à-dire que, en cas de doute, le logiciel doit s'abstenir d'intervenir. D'autant plus que le RFC rappelle que tous les lecteurs doivent voir le même article, ce qui interdit les modifications qui ne soient pas purement techniques.

Et il y a des règles plus pratiques. Par exemple, la section 3.2 est consacrée au champ `Path:` des entêtes. Il sert à indiquer le chemin parcouru par le message entre les "*relaying agent*" et donc à éviter les boucles. Tout serveur doit donc indiquer son identité dans ce champ (section 3.1.5 du RFC 5536, l'identité est en général le nom de domaine du serveur). Les sections 3.2.1 et 3.2.2 détaillent le processus de formation du champ `Path:`, bien plus riche que dans le RFC 1036. Un `Path:` peut être aussi complexe que :

```
Path: foo.isp.example!.SEEN.isp.example!!foo-news
           !.MISMATCH.2001:DB:0:0:8:800:200C:417A!bar.isp.example
           !!old.site.example!barbaz!!baz.isp.example
           !.POSTED.dialup123.baz.isp.example!not-for-mail
```

qui utilise la plupart des possibilités comme celle de noter que la machine `2001:DB::8:800:200C:417A` n'avait peut-être pas le droit de se prétendre `bar.isp.example`.

Tout aussi important que `Path:` qui permet d'éviter les boucles et de détecter les problèmes, par exemple les injections de spam, est l'en-tête `Message-ID:` (section 3.1.3 du RFC 5536) qui sert à assurer l'unicité des articles et à garantir que l'algorithme d'inondation se termine : un serveur refuse les articles qu'il a déjà. Pour s'en souvenir, il doit donc stocker les `Message-ID:` dans une base de données, souvent nommée `history` (section 3.3).

La section 3.4 se penche ensuite sur le cas des "*posting agents*", le logiciel avec lequel l'utilisateur va interagir pour écrire sur les "*news*". Parmi leurs nombreuses tâches figure celles liées aux réponses à des messages existants (sections 3.4.3 et 3.4.4). Le logiciel doit notamment construire un champ `References:` qui citera le `Message-ID:` du message auquel on répond, permettant ainsi au lecteur de "*news*" de reconstituer les fils de discussion (même s'il y aura toujours des ignorants pour voler les fils `<http://www.bortzmeyer.org/ne-pas-voler-les-fils.html>`, comme avec le courrier). Voici la partie de l'en-tête qui montre une réponse au message `<l_CdnfgkWe7PWyLUnZ2dnUVZ8viWnZ2d@giganews.com>` :

```
Newsgroups: fr.comp.reseaux.ethernet
References: <l_CdnfgkWe7PWyLUnZ2dnUVZ8viWnZ2d@giganews.com>
Subject: =?iso-8859-1?Q?Re:_Debogage_d'une_panne_r=E9seau?=
Message-ID: <49bfceec$0$2734$ba4acef3@news.orange.fr>
```

Les "*news*" ne sont pas seulement une technique mais avant tout un réseau social (même si le mot n'était pas encore à la mode lors de leur création). Certains groupes sont donc modérés et il faut donc aussi spécifier les règles techniques que doivent observer les modérateurs (section 3.9).

Il n'y a pas que les "*news*" dans la vie et certains groupes peuvent intéresser des gens qui ne veulent ou ne peuvent pas accéder aux "*news*". Le rôle des passerelles est donc de convertir les articles depuis, ou vers le monde des "*news*". Par exemple, une passerelle peut traduire les "*news*" d'un groupe en HTML et les mettre sur une page Web (voyez par exemple l'archive de comp.os.research <ftp://ftp.cse.ucsc.edu/pub/comp.os.research/>). Mais les passerelles les plus répandues sont celles entre les "*news*" et le courrier électronique, permettant aux utilisateurs du courrier d'écrire sur les "*news*" ou bien de recevoir des "*news*" par courrier. Historiquement, les passerelles ont souvent été responsables de problèmes sur les "*news*", par exemple de boucles sans fin (article transmis du réseau A au réseau B puis retransmis vers A et ainsi de suite éternellement) et la section 3.10 normalise donc ce qu'elles doivent faire et ne pas faire. Les passerelles doivent notamment conserver le `Message-ID:`, principale protection contre les boucles, puisque permettant de voir qu'un message est déjà passé. La section 3.10.4 donne un exemple complet pour le cas d'une passerelle bidirectionnelle avec le courrier.

Aujourd'hui, la plupart des "*news*" sont transportées par le protocole NNTP, dont le RFC 3977 date d'un peu plus de deux ans. Mais, historiquement, UUCP (RFC 976) avait été très utilisé. La section 2 du RFC détaille les obligations de ces protocoles de transport, notamment le fait de pouvoir faire passer des caractères codés sur 8 bits (permettant ainsi des encodages comme Latin-1). Cette obligation, nécessaire à l'internationalisation a fait l'objet de longues luttes, depuis l'époque où les logiciels devaient encoder les messages écrits en français avec le "*Quoted-Printable*". L'obligation s'applique aussi aux en-têtes des messages et un logiciel de "*news*" a donc davantage d'obligations ici qu'un logiciel de messagerie.

Une des particularités des "*news*" est que les messages de contrôle (comme ceux demandant la création ou la suppression d'un groupe) sont des messages comme les autres, distribués par le même mécanisme d'inondation. La section 5 normalise ces messages. Le principe de base est que le message de contrôle se reconnait par la présence d'un champ `Control:` (section 3.2.3 du RFC 5536). Dans l'ancien RFC 1036, les messages de contrôle pouvaient aussi se reconnaitre par un sujet spécial, commençant par `cmsg` mais cet usage est désormais officiellement abandonné.

Il n'y a aucune sécurité dans les "*news*" en général. Certains réseaux peuvent ajouter leurs propres mécanismes mais Usenet, par exemple, n'a pas de mécanisme général. Les messages de contrôle peuvent donc facilement être imités, même s'il existe des méthodes non officielles pour les authentifier <ftp://ftp.isc.org/pub/pgpcontrol/FORMAT> (section 5.1). Usenet étant un réseau très animé et très peu policé, cette absence de sécurité a donc entrainé souvent des surprises. Le RFC rappelle donc l'importance de tenter de s'assurer de l'authenticité d'un message de contrôle (les méthodes pour cela ne sont pas normalisées mais, par exemple, INN peut vérifier les signatures PGP <http://www.eyrie.org/~eagle/software/inn/docs/pgpverify.html>).

Les messages de contrôle permettent, par exemple, d'annuler un message, de stopper sa diffusion (section 5.3). Comme on s'en doute, ce sont parmi les messages les plus souvent usurpés. Le RFC 1036 demandait, dans l'espoir de renforcer la sécurité, que l'expéditeur (tel qu'indiqué par le champ `From:`) soit le même dans le message de contrôle et dans le message initial. Comme il est trivial d'usurper cette identité, cela ne faisait que diminuer la traçabilité, en décourageant les « annulateurs » de s'identifier. Cette règle est donc supprimée par notre RFC.

Les messages de contrôle liés à la gestion des groupes sont décrits dans la section 5.2. Par exemple, `newgroup` (section 5.2.1) permet de créer un nouveau groupe. Il ressemble à :

```
From: "example.* Administrator" <admin@noc.example>
Newsgroups: example.admin.info
Control: newgroup example.admin.info moderated
...
```

Parlant de sécurité, la section 6 lui est entièrement consacrée. Elle résume dès le début la situation : "*NetNews*" avait été conçu pour la diffusion large de l'information, et la sécurité en est presque totalement absente. Des grosses erreurs de mise en œuvre avaient en outre aggravé les choses comme le fait que certains logiciels appliquaient (il y a longtemps) les messages de contrôle... en les passant, verbatim, au shell !

Avant de participer aux "*news*" il faut donc bien être conscient ce ces problèmes : vulnérabilité aux dénis de service (section 6.2), non-authentification de l'expéditeur (attention donc si un message semble particulièrement outrancier, son vrai expéditeur n'est peut-être pas celui affiché dans le champ `From:`) et diffusion très large, parfois au delà de ce qui était prévu (section 6.3).

Les articles de "*news*" étaient traditionnellement marqués, dans le système MIME comme `message/news`. Ce type n'a jamais été un grand succès et notre RFC 5537 le remplace par trois nouveaux types, décrits en section 4, `application/news-transmission` pour un articles en route vers un "*injecting agent*" ou bien un modérateur et deux types pour les messages de contrôle, `application/news-groupinfo` et `application/news-checkgroups`.

L'annexe A résume les changements depuis le RFC 1036. Parmi eux :

— Un format beaucoup plus riche du champ `Path:`, permettant de mettre davantage d'informations comme la notification de suspicions sur l'origine d'un message,
— Les nouveaux types MIME de la section 4,
— Les nouvelles règles pour le format des messages de contrôle, notamment la suppression de la règle du sujet commençant par `cmsg`,
— La suppression de la règle de pseudo-authentification des messages d'annulation (qui obligeait à ce que le champ `From:` du message et de l'annulation correspondent),
— Et de nombreuses précisions ou des normalisations de pratiques que tout le monde faisait mais qui n'étaient pas dans le précédent RFC.

Il existe aujourd'hui de nombreuses mises en œuvre des "*news*" en logiciel libre. Par example, parmi les logiciels client (lecture et écriture), xrn <`http://www.mit.edu/people/jik/software/xrn.html`> ou Thunderbird (ce dernier ne sert donc pas qu'au courrier). Parmi les serveurs, INN. Personnellement, je regrette que mon lecteur de courrier habituel, mutt, n'aie pas de fonction de lecture des "*news*", même si plusieurs modifications non officielles ont déjà été proposées.

En 2001, l'achat par Google de la société DejaNews a mis à la disposition du géant de la recherche sur Internet une archive de tous les messages Usenet depuis 1981. Cette archive est désormais cherchable par date sur Google Groups <`http://groups.google.com/advanced_search`>. Si vous voulez chercher mon premier article envoyé <`http://groups.google.com/group/comp.lang.ada/msg/00dfd9a2c477fb49`> sur les "*news*"... (Avec une double signature, une erreur de débutant.) Ou bien une discussion en 1993 sur le DNS <`http://groups.google.com/group/comp.protocols.tcp-ip.domains/msg/04609bad0e151c8a`>...