

RFC 7826 : Real-Time Streaming Protocol Version 2.0

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 décembre 2016

Date de publication du RFC : Décembre 2016

<http://www.bortzmeyer.org/7826.html>

Voici la version 2 du protocole bien connu RTSP, protocole servant à accéder à des flux vidéo. Comme c'est ce que j'utilise pour regarder la télévision sur l'écran de mon PC, je suis ravi que l'IETF se préoccupe de l'améliorer.

Comme beaucoup de protocoles dans le monde du multimédia (SIP, par exemple), RTSP est en fait uniquement un protocole de **contrôle**, permettant de déclencher ou d'arrêter des flux audio ou vidéo. Ces flux peuvent être temps-réel ou bien avoir simplement été stockés sur le disque d'un serveur. Donc, RTSP est une zapette logicielle. RTSP fait le contrôle et plusieurs protocoles peuvent être utilisés pour le transport des données, UDP, TCP, RTP, etc. À noter la taille impressionnante de ce RFC, avec plus de 300 pages. Ce n'est pas que le protocole soit si compliqué que cela, mais il y a beaucoup d'options et de choix.

La section 2 du RFC résume le protocole : RTSP est client/serveur, le client RTSP se connecte au serveur, un certain nombre de choix techniques sont faits et ensuite l'envoi des données commence. Physiquement, les messages sont du texte (la syntaxe de RTSP ressemble beaucoup à celle d'HTTP) bien que du binaire soit parfois possible. La ressource convoitée est identifiée par un URI de plan `rtsp` (ou `rtsp`s pour TLS) et cet URI contient le nom de la machine qui sera utilisée comme serveur. Par exemple, si je dis à mon logiciel RTSP d'utiliser `rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=658&flavour=ld`, la connexion RTSP sur TCP (ou TCP avec TLS) se fera avec `mafreebox.freebox.fr`. La requête RTSP inclut un certain nombre d'en-têtes comme dans HTTP, et parfois un corps (toujours comme en HTTP). Voici un exemple avec le client VLC. Je le lance avec `vlc 'rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=897'` et on voit (tcpdump ne sait pas apparemment décoder le RTSP mais Wireshark y arrive très bien) :

```
Internet Protocol Version 4, Src: 192.168.2.1 (192.168.2.1), Dst: 212.27.38.253 (212.27.38.253)
Transmission Control Protocol, Src Port: 45854 (45854), Dst Port: rtsp (554), Seq: 563, Ack: 873, Len: 204
Real Time Streaming Protocol
  Request: PLAY rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=897 RTSP/1.0\r\n
  CSeq: 5\r\n
  User-Agent: LibVLC/2.0.3 (LIVE555 Streaming Media v2012.05.17)\r\n
  Session: pokf6CQWbA8CUyC
  Range: npt=0.000-\r\n
  \r\n
```

Dans l'exemple ci-dessus, le protocole était RTSP version 1.0 (rappelez-vous que ce RFC décrit la version 2), la requête était `PLAY` (dont le nom dit bien ce qu'elle fait et vous ne serez pas surpris d'apprendre qu'il existe une commande `PAUSE`) et l'un des en-têtes, `User-Agent` : montre que j'utilise bien `vlc`.

Quand au trafic lui-même, on voit (ici avec `tcpdump`) d'abord du RTSP sur TCP puis un gros flux UDP :

```
21:34:36.179830 IP (tos 0x10, ttl 64, id 20888, offset 0, flags [DF], proto UDP (17), length 1356)
    212.27.38.253.46099 > 192.168.2.1.34324: [udp sum ok] UDP, length 1328
21:34:36.180040 IP (tos 0x10, ttl 64, id 20889, offset 0, flags [DF], proto UDP (17), length 1356)
    212.27.38.253.46099 > 192.168.2.1.34324: [udp sum ok] UDP, length 1328
21:34:36.180738 IP (tos 0x10, ttl 64, id 20890, offset 0, flags [DF], proto UDP (17), length 1356)
    212.27.38.253.46099 > 192.168.2.1.34324: [udp sum ok] UDP, length 1328
```

Les contenus auxquels on accède avec RTSP peuvent être de type très variés. Il faut donc une description formalisée des caractéristiques de ce contenu. RTSP peut utiliser plusieurs formats pour cela, le plus répandu étant sans doute SDP (RFC 4566¹). C'est en tout cas celui utilisé entre mon VLC et ma Freebox. La description peut inclure le nombre de flux (souvent un flux vidéo et plusieurs audios), le protocole de délivrance (RTP - RFC 3550 - dans l'exemple ci-dessous), le format (MPEG-2 ici), etc :

```
Session Description Protocol
  Session Description Protocol Version (v): 0
  Owner/Creator, Session Id (o): leCDN 1395332443 1395332443 IN IP4 kapoueh.proxad.net
...
  Media Description, name and address (m): video 0 RTP/AVP 33
    Media Type: video
    Media Port: 0
    Media Protocol: RTP/AVP
    Media Format: MPEG-II transport streams
  Media Attribute (a): control:rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=658&f
  Media Attribute Fieldname: control
  Media Attribute Value: rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=658&flav
```

Quels sont les changements par rapport à RTSP version 1, la version du RFC 2326 ? Les deux versions, quoique identiques dans leurs principes, ne sont pas compatibles (par exemple, la commande `PLAY` n'a plus le même comportement, des en-têtes ont changé de syntaxe sans changer de nom, etc). C'est toujours un choix difficile que de casser la compatibilité d'un protocole mais, là, c'était nécessaire vu le nombre de modifications. En outre, RTSP 1 ne permettait pas de déployer facilement des extensions (en-têtes à la syntaxe trop rigide) et le modèle d'extension a changé. L'annexe I de notre RFC résume ce qu'il faut savoir sur ces différences : suppression des requêtes `RECORD` et `ANNOUNCE`, suppression de l'option qui permettait de faire passer RTSP (le contrôle, pas les données) sur UDP, gestion complète d'IPv6 (qui manquait en version 1), refactorisation du RFC (les en-têtes qui sont proches de ceux de HTTP sont désormais décrits par un texte spécifique, sans renvoyer au RFC HTTP), etc.

Il y a apparemment au moins une mise en œuvre de RTSP qui a la version 2, et plusieurs des nouveautés de la version 2 ont été mises en œuvre de manière séparée.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc4566.txt>