

FreeBox multiposte : Une solution pour un système Unix derrière un routeur / pare-feu

Peter FELECAN

28 janvier 2007

Table des matières

1	Introduction	1
2	Environnement matériel et logiciel	1
3	Des ports mouvants	1
3.1	Plongée verbeuse	2
3.2	Translation de ports	4
3.3	Replonger et stabiliser	4
4	La solution	6
4.1	Validation	7
4.2	Finalisation	8
5	Utiliser VLC	9
6	Conclusion	9

1 Introduction

Rien de plus simple que de regarder et enregistrer sur son ordinateur des émissions de télévision dans le bouquet offert par Free à ses abonnés dégroupés. Tout au moins si l'installation que nous possédons est simple.

Pour ceux qui, comme moi, ont un réseau local peuplé de systèmes fonctionnant sous Unix, un routeur filtrant et translatant les adresses, les chose se corsent un peu.

Nous pouvons trouver des solutions plus ou moins complètes sur les forums de discussions et sur quelque sites dans la sphère d'influence de Free, mais la recherche n'est pas toujours satisfaisante ou de tout repos.

Après moult expérimentations, j'ai trouvé un solution qui me donne entière satisfaction. Elle n'a rien d'original, tout est dit, de façon morcelée, ailleurs. La seule motivation pour son exposition est de réunir en un seul endroit tous les éléments dans l'espoir qu'elle soit utile à d'autres.

2 Environnement matériel et logiciel

Je suis un abonné dégroupé de Free. Assez loin du NRA mais avec un ligne saine et propre qui me permet d'avoir un accès de bonne qualité à une seule chaîne de télévision à la fois.

Ma FreeBox V5 est en mode *bridge* et les seuls équipements connectés sont la FreeBox HD par MIMO et un routeur, par Ethernet. Le routeur assure la translation d'adresses — NAT — et

ne laisse entrer rien sur son interface externe, coté Free ; sur l'interface interne, coté réseau local, tout peut sortir.

L'ensemble des systèmes connectés au réseau local fonctionnent sous différentes incantation d'Unix — Solaris, SPARC et X86 et Debian Etch, X86 et POWERPC.

Mon couteau suisse pour la vidéo est MPlayer. La version utilisée est 1.0rc1.

3 Des ports mouvants

Mon premier essai à été simplissime. Pour visionner **France Télévision 2** j'ai utilisé :

```
$ mplayer rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
```

Après quelques messages classiques, le programme est en attente de la réponse du serveur :

```
Playing rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201.
Resolving mafeebox.freebox.fr for AF_INET6...
Couldn't resolve name for AF_INET6: mafeebox.freebox.fr
Resolving mafeebox.freebox.fr for AF_INET...
Connecting to server mafeebox.freebox.fr[212.27.38.253]: 554...
bind: Cannot assign requested address
rtsp_session: unsupported RTSP server. Server type is 'fbxrtspd/1.1 Freebox minimal RTSP
STREAM_LIVE555, URL: rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
Stream not seekable!
file format detected.
Initiated "video/MP2T" RTP subsession on port 32770
```

Et rien d'utile ou intéressant ne se passe.

Plusieurs tentatives plus loin, nous pouvons observer que la valeur du port apparaissant dans la dernière ligne change, et prend des valeurs comme 32770, 32771, 32772.

3.1 Plongée verbeuse

Pour comprendre ce qui se passe, nous augmentons le niveau de verbosité du programme et diriger sa sortie standard et des erreurs vers un fichier que nous pourrions consulter à loisir :

```
$ mplayer -V rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201 2>&1 | tee /tmp/mp.log
```

Voici l'analyse sommaire du résultat obtenu — se référer au points correspondants dans la sortie du programme donné ci-dessous :

1. La connexion au serveur se fait correctement et les informations données sur les lignes suivante confirment cette analyse — il s'agit des éléments du SDP *Session Description Protocol*.
2. Ici nous trouvons l'ensemble de ports utilisés pour la communication entre le serveur et le client : 31336, 31337 et, respectivement 32768, 32769.
3. Voici la ligne que nous avons vu dans l'utilisation non verbeuse du programme ; la valeur du port a changée de nouveau !
4. Cette ligne donne un résumé des paramètres qui nous intéressent.

```
Playing rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201.
get_path('sub/') -> '/home/peter/.mplayer/sub/'
STREAM_RTSP, URL: rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
Filename for url is now rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
Filename for url is now rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
Resolving mafeebox.freebox.fr for AF_INET6...
```

```

    Couldn't resolve name for AF_INET6: mafreebox.freebox.fr
    Resolving mafreebox.freebox.fr for AF_INET...
(1) Connecting to server mafreebox.freebox.fr[212.27.38.253]: 554...
(1) SDP:
(1) v=0
(1) o=leCDN 1169979463 1169979463 IN IP4 kapoueh.proxad.net
(1) s=France 2
(1) i=France 2
(1) c=IN IP4 0.0.0.0
(1) t=0 0
(1) m=video 0 RTP/AVP 33
(1) a=control:rtsp://212.27.38.253/freeboxtv/stream?id=201

(2) RTP Port from SDP appears to be: 31336
(2) RTCP Port from SDP appears to be: 31337
(2) RTSP Transport: Transport: RTP/AVP;unicast;client_port=31336-31337;mode="PLAY"
(2) RTSP Destination: 212.27.38.254
(2) Client RTP port : 31336
(2) Client RTCP port : 31337
(2) Server RTP port : 32768
(2) Server RTCP port : 32769
    bind: Cannot assign requested address
    RTP Sock : -1
    RTCP Sock : 5
    rtsp_session: unsupported RTSP server. Server type is 'fbxrtspd/1.1 Freebox minimal
    Filename for url is now rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
    Filename for url is now rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
    STREAM_LIVE555, URL: rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
    STREAM: [RTSP and SIP] rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201
    STREAM: Description: standard RTSP and SIP
    STREAM: Author: Ross Finlayson
    STREAM: Comment: Uses LIVE555 Streaming Media library.
    Stream not seekable!
    file format detected.
    Sending request: DESCRIBE rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201 RTSP/
    CSeq: 1
    Accept: application/sdp
    User-Agent: MPlayer (LIVE555 Streaming Media v2006.03.16)

Received DESCRIBE response: RTSP/1.0 200 OK
Cseq: 1
Content-Length: 181
Content-Type: application/sdp
Content-Language: fr
Server: fbxrtspd/1.1 Freebox minimal RTSP server

Need to read 181 extra bytes
Read 181 extra bytes: v=0
o=leCDN 1169979463 1169979463 IN IP4 kapoueh.proxad.net
s=France 2
i=France 2
c=IN IP4 0.0.0.0

```

```

t=0 0
m=video 0 RTP/AVP 33
a=control:rtsp://212.27.38.253/freeboxtv/stream?id=201

(3) Initiated "video/MP2T" RTP subsession on port 32774
Increased video socket receive buffer to 2000000 bytes
Sending request: SETUP rtsp://212.27.38.253/freeboxtv/stream?id=201 RTSP/1.0
CSeq: 2
Transport: RTP/AVP;unicast;client_port=32774-32775
User-Agent: MPlayer (LIVE555 Streaming Media v2006.03.16)

Received SETUP response: RTSP/1.0 200 OK
Cseq: 2
Session: zWwg6sJfciXuQ7yi62I5IpwW0qU8foiEbFL8Xkd0t0k9YJiVBRQa6cWWtHVzW44
Server: fbxrtspd/1.1 Freebox minimal RTSP server

(4) Transport: RTP/AVP;unicast;mode=play;destination=212.27.38.254;client_port=32774-32775

Sending request: PLAY rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201 RTSP/1.0
CSeq: 3
Session: zWwg6sJfciXuQ7yi62I5IpwW0qU8foiEbFL8Xkd0t0k9YJiVBRQa6cWWtHVzW44
Range: npt=0.000-
User-Agent: MPlayer (LIVE555 Streaming Media v2006.03.16)

Received PLAY response: RTSP/1.0 200 OK
Cseq: 3
Session: zWwg6sJfciXuQ7yi62I5IpwW0qU8foiEbFL8Xkd0t0k9YJiVBRQa6cWWtHVzW44
Server: fbxrtspd/1.1 Freebox minimal RTSP server

==> Found video stream: 0

```

L'hypothèse qui s'impose, au vu de la session ci-dessus et de la topologie du réseau — routeur faisant du NAT —, est que nous avons un problème de translation de ports.

3.2 Translation de ports

Pour vérifier cette hypothèse, une exploration du journal du routeur s'avère utile ; voici l'extrait significatif :

```

IP entry duplicated 1623 times
3583441 sec IP Packet discarded from 212.27.38.253 port 59813 to 8x.2xx.x.1xx port 32774
IP entry duplicated 1 times
3583441 sec IP Packet discarded from 212.27.38.253 port 59813 to 8x.2xx.x.1xx port 31336

```

En effet, les port 32774 et 31336 font partie de ceux utilisés pour communiquer entre le serveur Free et le programme MPlayer, dans le sens *entrant*.

La validation de l'hypothèse peut se faire en ouvrant ces ports dans les règles de translation de ports du routeur pour qu'ils soient envoyé au système utilisant MPlayer.

Toutefois, il reste une difficulté : quels ports ouvrir ? Nous avons vu que leur valeur n'est pas constante et il n'est pas question d'ouvrir plus que le nécessaire et suffisant !

3.3 Replonger et stabiliser

Avant de valider notre hypothèse, nous devons voir quels sont les ports variables.
Pour cela, un petit programme :

```
#!/bin/bash

iteration=1
while [ "${iteration}" -le 10 ]
do
{
printf "\niteration %d\n" "${iteration}"
(mplayer \
-V \
rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201 2>&1 | \
egrep -e '^Client ' -e '^Server ' -e 'video.MP2T')&
sleep 60
pkill mplayer
((iteration = iteration + 1))
}
done

exit 0
```

nous permet de voir l'information qui nous intéresse :

```
iteration 1
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 32778

iteration 2
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 32780

iteration 3
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 32780

iteration 4
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 32782

iteration 5
```

Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 32784

iteration 6
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 32786

iteration 7
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 32788

iteration 8
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 32790

iteration 9
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 32792

iteration 10
Client RTP port : 31336
Client RTCP port : 31337
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 32794

Nous pouvons constater que la valeur des ports pour communiquer entre le serveur et le client — pour l'information *entrante* — restent constantes :

Client RTP port : 31336
Client RTCP port : 31337

La seule valeur de port entrant fluctuante est celle du port utilisé pour la sous-session video/MP2T.
À noter que les ports *sortants* fluctuent aussi :

Server RTP port : 32768
Server RTCP port : 32769

dans les itérations 1, 3, 5, 7 et 9

Server RTP port : 32770
Server RTCP port : 32771

dans les itérations 2, 4, 6, 8 et 10.

Toutefois, n'ayant pas de restriction de sortie dans nos règles de filtrage, ces valeurs ne nous intéressent pas.

4 La solution

Pour stabiliser les ports d'entrants, nous allons forcer MPlayer à utiliser toujours les mêmes ports en donnant l'option `-rtsp-port 44122`¹. Ainsi, les ports 44122 et 44123 seront utilisés pour la communication *entrante*.

4.1 Validation

Une version légèrement modifié du programme présentée dans la section précédente est utilisé pour valider notre hypothèse :

```
#!/bin/bash

iteration=1
while [ "${iteration}" -le 10 ]
do
{
printf "\niteration %d\n" "${iteration}"
(mplayer \
-V \
-rtsp-port 44122 \
rtsp://mafreebox.freebox.fr/freeboxtv/stream?id=201 2>&1 |
egrep -e '^Client ' -e '^Server ' -e 'video.MP2T')&
sleep 60
pkill mplayer
((iteration = iteration + 1))
}
done

exit 0
```

Notez que nous choisissons un numéro de port non affecté par IANA.

Le résultat :

```
iteration 1
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 44122

iteration 2
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 44122
```

¹L'extrait du manuel qui nous intéresse est : `-rtsp-port` Used with `'rtsp://'` URLs to force the client's port number. This option may be useful if you are behind a router and want to forward the RTSP stream from the server to a specific client.

iteration 3
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 44122

iteration 4
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 44122

iteration 5
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 44122

iteration 6
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 44122

iteration 7
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 44122

iteration 8
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32770
Server RTCP port : 32771
Initiated "video/MP2T" RTP subsession on port 44122

iteration 9
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32768
Server RTCP port : 32769
Initiated "video/MP2T" RTP subsession on port 44122

iteration 10
Client RTP port : 44122
Client RTCP port : 44123
Server RTP port : 32770
Server RTCP port : 32771

Initiated "video/MP2T" RTP subsession on port 44122

nous montre que nous obtenons une stabilisation des ports.
Seuls les ports 44122 et 44123 sont utilisé pour les informations entrantes.

4.2 Finalisation

Pour finaliser la solution, nous devons définir une règle de translation de port.

Il s'agit d'une règle *entrante*, pour le protocole UDP, sur les ports 44122 et 44123 vers l'adresse locale du système utilisé pour regarder ou enregistrer le flux télévisuel.

Dans le cas du routeur utilisé, cette règle s'exprime ainsi :

```
Forward UDP, port range from port 44122 to port 44123
To Trusted Address 192.168.xxx.xxx
Name for this service freetv
```

5 Utiliser VLC

Nous avons vu ci-dessus comment utiliser MPlayer. Beaucoup d'utilisateurs préfèrent une interface homme-machine graphique à une interface homme-machine en ligne de commande. Question de goût...

Voici une solution rapide, dans les même conditions que celles évoquées dans les sections précédentes, sans utiliser une version spécialement modifiée du logiciel.

La version testé est la 0.8.6.

La modification est à faire dans le paramétrage du logiciel :

```
Settings
├─ Preferences
│   └─ Input / Codecs
│       └─ Demuxers
│           └─ RTP / RTSP (choisir Advanced Options)
│               └─ Client Port : 44123
```

Les ports ainsi utilisés seront 44122 et 44123. Observez comment VLC utilise le port donné : à l'inverse de MPlayer.

6 Conclusion

Nous pouvons regarder et enregistrer les flux télévisuels du bouquet Free sous Unix en utilisant un routeur faisant de la translation d'adresses et de ports.

Les 2 logiciels testés sont MPlayer et VLC, sans aucune modification du code.

Contrairement à ce qui est dit sur de nombreux forums de discussion, il n'est pas nécessaire d'ouvrir son pare-feu plus que nécessaire et le seul problème est posé par la variabilité du port utilisé par la sous-session entrante `video/MP2T`.

Pour ceux qui ont des règles pour les communications sortantes, il faut ajouter une règle pour le protocole TCP, le port 554, et la plage de ports 32768 à 32771 pour le protocole UDP. Notez, que je n'ai pas testé cette configuration ; il s'agit juste d'une extrapolation théorique.