

Configurer les ports eMule ou mDonkey pour un routeur Linux

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 août 2007. Dernière mise à jour le 8 août 2007

<https://www.bortzmeyer.org/emule-ports-linux.html>

Le logiciel de partage de fichiers eMule, s'il n'est pas contactable par d'autres utilisateurs depuis l'extérieur, rend un service dégradé. Comment le rendre contactable si le routeur est une machine Linux ?

On sait que le NAT est une source d'ennuis sans fin, notamment pour les applications pair à pair. Sa présence nécessite beaucoup de bricolages comme ici, où nous allons faire du "*port forwarding*", c'est-à-dire relayer les paquets IP depuis un pair vers notre client eMule ou mDonkey, via un routeur Linux.

Le cas d'un routeur Linux n'est pas actuellement documenté en <http://www.emule-project.net/home/perl/help.cgi?l=1&rm=show_entries&cat_id=251> mais, par contre, il y a un exemple dans l'excellente documentation de mDonkey <<http://mldonkey.sourceforge.net/WhatFirewallPortsToOpen>>. Mais c'est de toute façon assez simple. Il faut commencer par connaître les ports utilisés. eMule en change désormais, pour passer les filtrages. Allez dans Préférences puis Connexion et regardez "Port client". Notez la valeur pour le port TCP et le port UDP.

Pour mDonkey, on peut utiliser la commande `portinfo` de la console :

```
> portinfo
--Portinfo--
Network  | Port|Type
-----+-----+-----
BitTorrent| 6882|client_port TCP
BitTorrent| 6881|tracker_port TCP
Core      | 4080|http_port
Core      | 4000|telnet_port
Core      | 4001|gui_port
Donkey    | 4550|client_port TCP
Donkey    | 4554|client_port UDP
Donkey    | 18703|overnet_port TCP+UDP
Donkey    | 10349|kademlia_port UDP
G2        | 6347|client_port TCP+UDP
Gnutella  | 6346|client_port TCP+UDP
```

ou bien un très bon script bash <<http://mldonkey.sourceforge.net/BashScriptToFindPortsUsed>>.

Mettons que le port TCP soit 4550 et le port UDP 4554. On configure le routeur ainsi avec iptables dans un petit script shell :

```
#!/bin/sh

# Valeurs spécifiques a mon site
# Ports utilisés par eMule / ml Donkey
tcp=4550
udp=4554
# Interface d'entrée
in_interface=ppp0
# La machine qui a eMule
emule=10.19.1.25

# TCP
iptables --table nat --append PREROUTING --protocol tcp \
  --in-interface ${in_interface} --dport ${tcp} \
  --jump DNAT --to ${emule}:${tcp}
iptables --insert FORWARD --protocol tcp --in-interface ${in_interface} \
  --dport ${tcp} \
  --jump ACCEPT

# UDP
iptables --table nat --append PREROUTING --protocol udp \
  --in-interface ${in_interface} --dport ${udp} --jump DNAT \
  --to ${emule}:${udp}
iptables --insert FORWARD --protocol udp --in-interface ${in_interface} \
  --dport ${udp} \
  --jump ACCEPT
```

Vous pouvez vérifier que les règles ont bien été insérées avec `iptables --table nat --list PREROUTING` et `iptables --list FORWARD`. L'option `-e` est également très pratique car elle affiche le nombre de paquets qui ont été acceptés par chaque règle. `tcpdump` ou `wireshark` sont également des moyens très pratiques de voir ce qui se passe sur le réseau, en cas de problèmes, ou tout simplement pour apprendre.

Pour être exécutées automatiquement au démarrage de la machine, ces instructions doivent être mises dans un fichier exécutable, placé dans le répertoire des scripts exécutés au démarrage. Par exemple, sur Debian, on les met dans un fichier nommé, mettons, `port-forwarding`, placé dans `/etc/init.d`, et on place les liens symboliques nécessaires avec une commande comme `sudo update-rc.d port-forwarding start 2 3 4 ..` Si l'interface vers l'Internet est une interface dynamique, par exemple avec PPP, on peut plutôt mettre le script dans le répertoire des commandes exécutées lorsque l'interface devient active, `/etc/ppp/ip-up.d`.

Une fois ce réglage fait, cliquez sur "Test des ports" dans la même fenêtre Préférences -> Connexion -> "Port client". Votre navigateur va se connecter à un service Web qui doit tester TCP et UDP, vous afficher qu'ils marchent, puis un message sympa du genre « Amusez-vous bien avec eMule » qui vous informe que vous n'êtes plus un membre de seconde classe, un méprisable "Low ID" <<http://mldonkey.sourceforge.net/LowID>>. Pour mlDonkey, vous pouvez regarder votre ID avec la commande console `id`:

```
> id
For 89.54.92.142:3333 () ---> waiting
For 80.239.200.101:3000 (BiG BanG 2) ---> 8212756 (LowID)
For 62.241.53.2:4242 (DonkeyServer No1) ---> 13844656 (LowID)
For 80.239.200.99:3000 (BiG BanG 1) ---> 10864435 (LowID)
```

Si on a bien activé la correspondance des ports, on doit avoir plutôt quelque chose comme :

```
For 67.159.44.187:4232 (www.wmule.com ) ---> 192.0.2.9 (HighID)
For 212.179.18.142:4232 (best hash ) ---> 192.0.2.9 (HighID)
For 67.159.44.107:4232 (www.wmule.com ) ---> 192.0.2.9 (HighID)
For 212.179.18.134:4232 (www.wmule.com ) ---> 192.0.2.9 (HighID)
```

Si on veut connaître tous les détails techniques, ils sont bien exposés en http://www.emule-project.net/home/perl/help.cgi?l=1&topic_id=122&rm=show_topic.