Mise en place d'un MLVPN

Qu'est-ce que MLVPN

MLVPN est une solution Open Source permettant de faire de l'agrégation de lien WAN entrant et sortant comme le propose OutTheBox de OVH

Installation de MLVPN

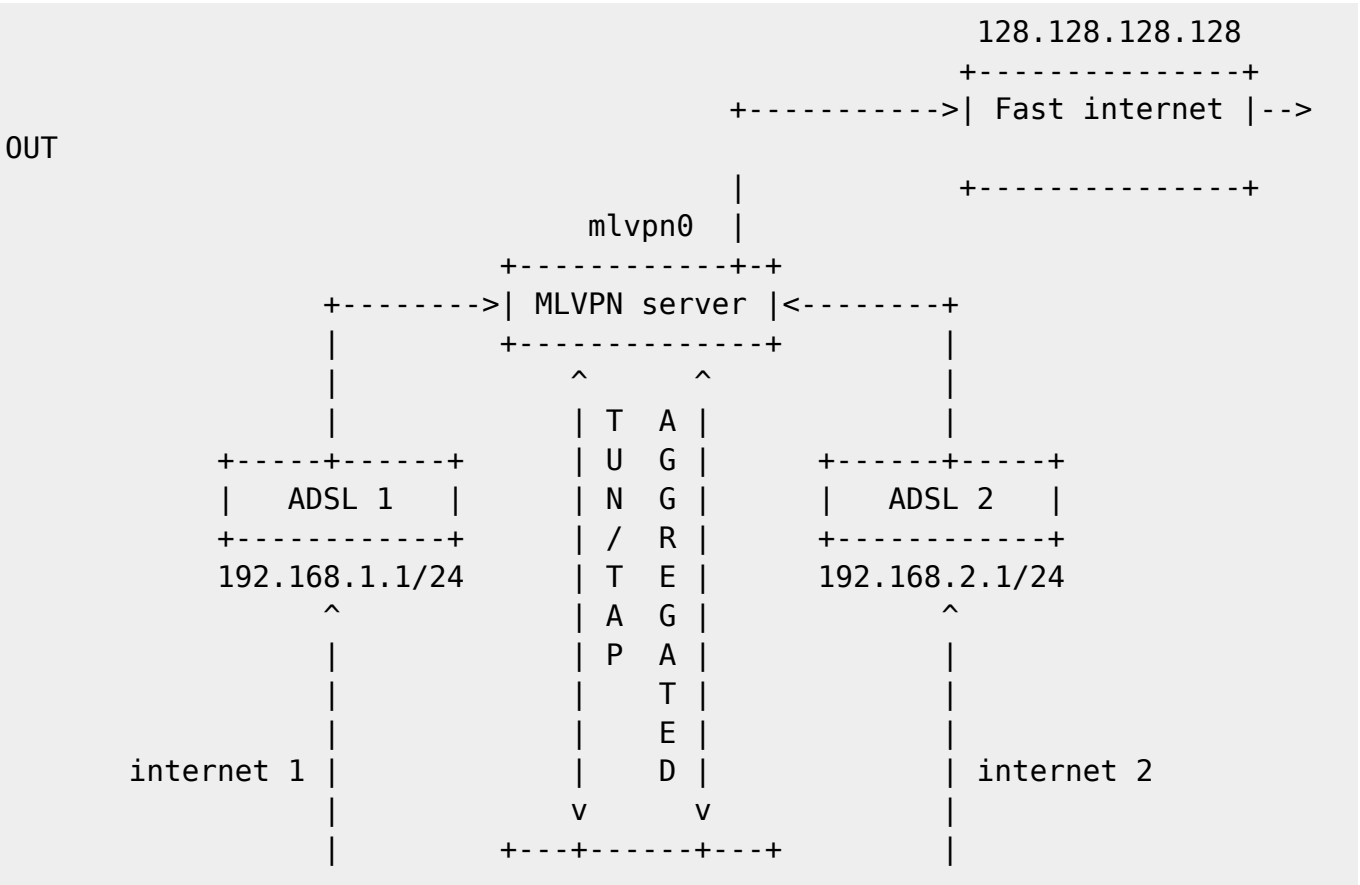
Debian Jessie/Sid

Voici les commandes necessaires pour installer MLVPN

```
apt-key adv --keyserver pgp.mit.edu --recv 3324C952
echo "deb http://debian.mlvpn.fr unstable/"
>/etc/apt/sources.list.d/mlvpn.list
apt-get update
apt-get install mlvpn
```

Configuration Standard

Voici le schéma sur lequel la configuration de base s'applique.



```
                  +---------| MLVPN client |---------+
                            +--------------+
                       mlvpn0 ; eth0: 192.168.0.1
                                       ^
   +------+                            |
   | LAN  |----------------------------+
   +------+
192.168.0.0/24
```

Côté serveur

Ce fichier est situé dans le dossier /etc/mlvpn/ et a pour nom “mlvpn.conf” par exemple

```
[general]
statuscommand = "/etc/mlvpn/mlvpn0_updown.sh"
tuntap = "tun"
mode = "server"
interface_name = "mlvpn0"
timeout = 30
password = "pleasechangeme!"
reorder_buffer_size = 64
loss_tolerence = 50

[filters]

[adsl1]
bindhost = 128.128.128.128
bindport = 5080

[adsl2]
bindhost = 128.128.128.128
bindport = 5081
```

Il est appeler un fichier dans l'option “statuscommand”, qui est un script. En voici un fonctionnel :

```
#!/bin/bash

error=0; trap "error=$((error|1))" ERR
tuntap_intf="$1"
newstatus="$2"
rtun="$3"
[ -z "$newstatus" ] && exit 1
(
if [ "$newstatus" = "tuntap_up" ]; then
    echo "$tuntap_intf setup"
    /sbin/ip link set dev $tuntap_intf mtu 1400 up
    # route vers le ou les réseaux du client
    /sbin/ip route add 192.168.0.0/24 dev $tuntap_intf
    # NAT dit de translation d'adresses, qui permet aux adresses du réseau
```

```
ci-dessous d’être NATé vers l'internet
    /sbin/iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.0/24 -j
MASQUERADE
elif [ "$newstatus" = "tuntap_down" ]; then
    # Suppression de la règle vu ci-dessus
    /sbin/iptables -t nat -D POSTROUTING -o eth0 -s 192.168.0.0/24 -j
MASQUERADE
fi
) >> /var/log/mlvpn_commands.log 2>&1
exit $errors
```

Ce script **DOIT** avoir pour droit 700 pour l'utilisateur root seulement.

Côté Client

Pour le client, des configurations de tables de routages iproute2 sont nécessaires (intégrer de base dans la distribution Debian).

Nous allons créer deux tables de routage supplémentaire, nommé dans l'exemple adsl1 avec l'id 101 et adsl2 avec l'id 102, grâce aux commandes :

```
root@mlvpnclient:~# echo 101 adsl1 >> /etc/iproute2/rt_tables
root@mlvpnclient:~# echo 102 adsl2 >> /etc/iproute2/rt_tables
```

Et donc ajouter des règles de routage dans les tables, afin de router les tunnels sur les WANs voulu. Il faudra scripter celui ci au démarrage pour automatiser le lancement du client.

```
# Route pour l'adsl1
ip route add 192.168.1.0/24 dev eth1 scope link table adsl1
ip route add default via 192.168.1.1 dev eth1 table adsl1

# Route pour l'adsl2
ip route add 192.168.2.0/24 dev eth2 scope link table adsl2
ip route add default via 192.168.2.1 dev eth2 table adsl2

# Cela va attribuer les tables en fonction du réseau source
ip rule add from 192.168.1.0/24 table adsl1
ip rule add from 192.168.2.0/24 table adsl2
```

On peux tester ces tables grâce aux commandes :

```
root@mlvpnclient:~# ping -I192.168.1.1 8.8.8.8
PING ping.ovh.net (213.186.33.13) 56(84) bytes of data.
64 bytes from 213.186.33.13: icmp_req=1 ttl=51 time=40.6 ms
64 bytes from 213.186.33.13: icmp_req=2 ttl=51 time=41.5 ms

root@mlvpnclient:~# ping -I192.168.2.1 8.8.8.8
PING ping.ovh.net (213.186.33.13) 56(84) bytes of data.
64 bytes from 213.186.33.13: icmp_req=1 ttl=51 time=62.0 ms
```

```
64 bytes from 213.186.33.13: icmp_req=2 ttl=51 time=64.1 ms
```

Au mettre titre que le serveur, il est nécessaire de mettre un fichier de configuration situé dans /etc/mlvpn/ avec pour nom en exemple mlvpn.conf

```
[general]
statuscommand = "/etc/mlvpn/mlvpn0_updown.sh"
tuntap = "tun"
mode = "client"
interface_name = "mlvpn0"
timeout = 30
password = "you have not changed me yet?"
reorder_buffer_size = 64
loss_tolerence = 50

[filters]

[adsl1]
bindhost = "192.168.1.2"
remotehost = "128.128.128.128"
remoteport = 5080

[adsl2]
bindhost = "192.168.2.2"
remotehost = "128.128.128.128"
remoteport = 5081
```

et comme pour le serveur, voici le script du “statuscommand” :

```
#!/bin/bash

error=0; trap "error=$((error|1))" ERR

tuntap_intf="$1"
newstatus="$2"
rtun="$3"

[ -z "$newstatus" ] && exit 1

(
if [ "$newstatus" = "tuntap_up" ]; then
    echo "$tuntap_intf setup"
    /sbin/ip link set dev $tuntap_intf mtu 1400 up
    /sbin/route del default dev eth1
    /sbin/route add default dev $tuntap_intf
elif [ "$newstatus" = "tuntap_down" ]; then
    echo "$tuntap_intf shutdown"
    /sbin/route del default dev $tuntap_intf
    /sbin/route add default dev eth1
elif [ "$newstatus" = "rtun_up" ]; then
    echo "rtun [${rtun}] is up"
```

```
elif [ "$newstatus" = "rtun_down" ]; then
    echo "rtun [${rtun}] is down"
fi
) >> /var/log/mlvpn_commands.log 2>&1

exit $errors
```

Il ne faut pas oublier les droits.

Lancement de MLVPN

Pour tester les configuration, il faut lancer la commande sur les deux partis de MLVPN.

```
root@server:~ # mlvpn --debug -v --user mlvpn -c /etc/mlvpn/mlvpn0.conf
```

Vous pourrez voir les logs dans /var/log/mlvpn_commands.log

Maintenant il faut tester le débit avec iperf ou avec un client.

Pour activer MLVPN au démarrage, il suffit de lancer la commande suivante sur les deux partis.

```
update-rc.d mlvpn enable
```

Notes :

En cas de sessions TCP qui ne fonctionnent pas, il faut ajouter la règle sous le client ou le serveur :

```
iptables -t mangle -A FORWARD -p tcp -m tcp --tcp-flags SYN,RST SYN -j
TCPMSS --clamp-mss-to-pmtu
```

From:
https://wiki.virtit.fr/ - **VirtIT**

Permanent link:
https://wiki.virtit.fr/doku.php/kb:linux:generalites:mise_en_place_d_un_mlvpn

Last update: **2018/01/18 16:20**