

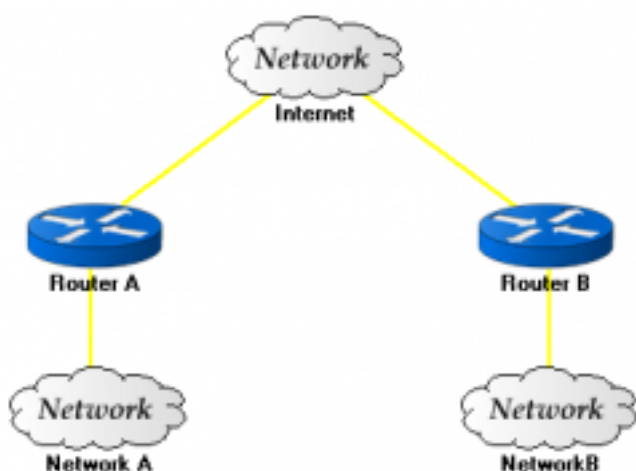


## MPD - использование в качестве pptp-клиента

Опубликовано muff в Ср, 2013-03-13 14:55

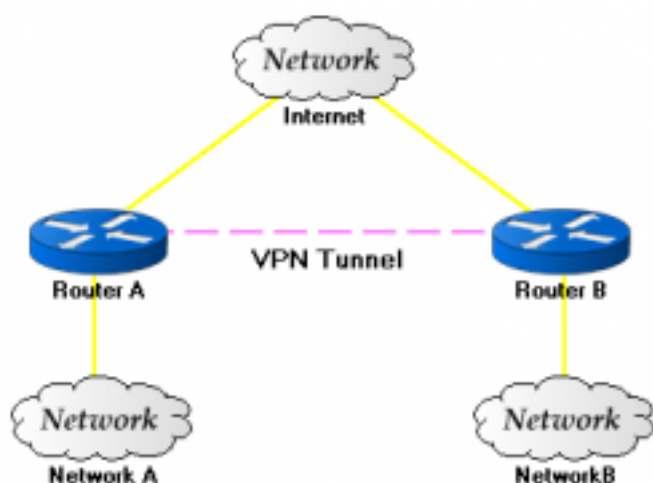
Понадобилось однажды ресурсы сети А транслировать как локальные ресурсы сети В. Поскольку прямого взаимодействия этих сетей нету, объединение сетей выполним посредством построения VPN-тунеля, благо что в роли маршрутизаторов обеих сетей выступают сервера под управлением **FreeBSD**. Поскольку в сети А уже функционировал полноценный [VPN-сервер на базе MPD](#) [1], то не будем изобретать велосипед, а просто настроим подключение с маршрутизатора сети В в качестве VPN-клиента к VPN-серверу, находящегося в сети А.

### Логика взаимодействия сетей А и В до построения VPN-тунеля



[2]

### Логика взаимодействия сетей А и В после построения VPN-тунеля



[3]

Пора приступать к настройке. Будем отталкиваться от того, что PPTP VPN-сервер на Router В уже настроен и функционирует. Соответственно, осталось только настроить PPTP VPN-клиент на Router А. В качестве VPN-клиента выбираем **MPD (Multi-link PPP daemon)**. Выполним установку **MPD5** из системы портов:

```
Router_A# cd /usr/ports/net/mpd5 && make install clean && rehash
```



По завершению установки приступаем к редактированию файла конфигурации **mpd.conf**. В результате нехитрых манипуляций, получаем следующий листинг **/usr/local/etc/mpd5/mpd.conf** (часть опций прокомментирована):

```
startup:          # ?????????? ?????????????? set user username username_pass
admin            set user username1 userpass1
# Конфигурация консоли set console self 127.0.0.1 5005 set console open # Ко
нфигурация веб-сервера set web self 0.0.0.0 5006 set web open

# ??????????, ??? ?????? ?????????? ? ?????? PPTP-????????default: load pptp_client

# ?????? ?????????????? PPTP-?????????
pptp_client:     create bundle static B1 # Этот скрипт обрабатывается при "поднятии" инте
рфейса set iface up-script /usr/local/etc/mpd5/up-script.sh
# Автоматическое добавление маршрута при "поднятии" интерфейса
set iface route 172.16.180.0/24 set ipcp ranges 0.0.0.0/0 0.0.0.0/0 create link static L1
pptp set link action bundle B1 # Параметры аутентификация на VPN-сервере
set auth authname "VPN_LOGIN" set auth password "VPN_PASSWD"
# Уменьшаем размер MTU для корректного прохождения трафика
set link mtu 1460 # Настройка автоподключения в случае обрыва сессии
set bundle no noretry set link max-redial 0 # Настройка keep-alive (поддерживание
сессии) set link keep-alive 20 75 # Указываем IP-адрес удаленного сервера
set pptp peer IP_VPN_SERVER_HERE
# Отключаем оконный механизм, назначаемый протоколом PPTP, # повышая этим пр
оизводительность. set pptp disable windowing open
```

где

- **username** - имя пользователя с правами администратора
- **username\_pass** - пароль пользователя с правами администратора
- **username1** - имя пользователя с привилегиями обычного пользователя
- **userpass1** - пароль пользователя с привилегиями обычного пользователя
- **VPN\_LOGIN** - имя пользователя для авторизации на VPN-сервере
- **VPN\_PASSWD** - пароль для авторизации пользователя на VPN-сервере
- **IP\_VPN\_SERVER\_HERE** - IP-адрес VPN-сервера
- **/usr/local/etc/mpd5/up-script.sh** - скрипт, который будет обрабатываться при старте MPD

В моем частном случае при старте MPD необходимо будет добавить только один маршрут, который я прописал прямо в **mpd.conf**. Если же необходимо через подключение получать маршрут по умолчанию, то необходимо модифицировать строку "**set iface route 178.162.180.0/29**" и заменить ее на "**set iface route default**".

Что интересно, при старте **MPD** после создания интерфейса, роут на IP-адрес VPN-сервера направляется в созданный туннель, соответственно маршрутизация работает некорректно. А после удаления маршрута все работает на ура, пинги ходят, маршруты доступны. Поэтому при запуске **MPD** будем удалять этот маршрут.

Соответственно, листинг скрипта **/usr/local/etc/mpd5/up-script.sh** будет следующий:

```
#!/bin/sh

/sbin/route delete IP_VPN_SERVER_HERE
```



Также необходимо для скрипта дать права на выполнение, чтобы он корректно обрабатывался:

```
# chmod +x /usr/local/etc/mpd5/up-script.sh
```

Добавляем загрузку **MPD** при старте системы:

```
# echo '# VPN PPTP Client' >> /etc/rc.conf
# echo 'mpd_enable="YES"' >> /etc/rc.conf
```

Перед первым запуском настроим логирование, чтобы можно было обнаружить ошибки конфигурации. Ну и не забываем о настройке ротации логов.

Настройка логирования:

```
# echo '!mpd' >> /etc/syslog.conf
# echo '*.* /var/log/mpd.log' >> /etc/syslog.conf
# touch /var/log/mpd.log
# killall -HUP syslogd
```

Настройка ротации логов:

```
# echo '/var/log/mpd.log 600 8 * $W6D0 JC' >> /etc/newsyslog.conf
```

После всех этих манипуляций запускаем **MPD**:

```
# sh /usr/local/etc/rc.d/mpd5 start
```

Проверяем, поднялся ли интерфейс:

```
# ifconfig ng0
```

```
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> metric 0 mtu 1460
inet 10.22.84.27 --> 10.221.84.1 netmask 0xffffffff
```

Собственно говоря - на этом все. Настройка **MPD5** как PPTP-клиента закончена. Дальнейшая настройка маршрутизации выходит за рамки данной статьи, поэтому этот этап упущен.

**P.S.** Протестировал пропускную способность PPTP-тунеля... [Iperf](#) [4] показал немногим больше 400 Mbps.

**Источник (получено 2026-07-06 09:10):**

<http://muff.kiev.ua/content/mpd-ispolzovanie-v-kachestve-pptp-klienta>

**Ссылки:**

[1] <http://muff.kiev.ua/content/mpd-nastroika-sobstvennogo-vpn-servera>

[2] [http://muff.kiev.ua/files/imagepicker/1/muff\\_vpn\\_cl01.png](http://muff.kiev.ua/files/imagepicker/1/muff_vpn_cl01.png)

[3] [http://muff.kiev.ua/files/imagepicker/1/muff\\_vpn\\_cl02.png](http://muff.kiev.ua/files/imagepicker/1/muff_vpn_cl02.png)

[4] <http://muff.kiev.ua/content/iperf-izmeryaem-propusknuyu-sposobnost-opornoj-seti>