



802.1Q VLAN - реализация 802.1Q VLAN на FreeBSD

Опубликовано muff в Ср, 2010-06-30 16:57

Уже довольно большой период времени все сетевые интерфейсы настраиваю с поддержкой **802.1Q VLAN**. Решил описать процедуру настройки, возможно мой опыт пригодится еще кому-либо.

Итак, не будем забывать о том, что есть сетевые карты, которые поддерживают `vlan` на аппаратном уровне, а есть которые не поддерживаются. Если сетевая карта не поддерживает `vlan`, во FreeBSD предусмотрена software поддержка, но карточка должна поддерживать oversized пакеты. Этим требованиям удовлетворяет большинство современных сетевых карт.

FreeBSD позволяет использовать `vlan`-интерфейсы двух типов:

- псевдоинтерфейс `vlan`
- cisco-like интерфейс

Псевдоинтерфейс `vlan`

Исходные данные:

- коммутатор настроен корректно
- порт коммутатора в режиме "trunk"
- VID 8 - Internet
- VID 12 - локальная сеть
- сетевой адаптер NVIDIA nForce MCP67 Networking Adapter (nfe0)

Для поддержки 802.1Q VLAN на уровне ядра, необходимо наличие в конфигурационном файле ядра наличие следующей строки:

```
device      vlan
```

Но если поддержки 802.1Q VLAN на уровне ядра нету, то ничего страшного, поддержка 802.1Q VLAN подгрузится в виде модуля:

```
# kldstat | grep vlan
```

```
4     1 0xffffffff80822000 2306      if_vlan.ko
```

Итак, приступим к созданию VID 8.

```
# ifconfig vlan8 create vlan 8 vlandev nfe0 195.3.159.250 netmask 255.255.255.252
```

Теперь более подробно о используемых опциях:

- **ifconfig** - команда конфигурирования сетевого интерфейса;
- **vlan8** - указываем, какой именно интерфейс будем конфигурировать (я имею привычку обозначать виртуальный интерфейс, отталкиваясь от VID, но ничего не мешает проводить нумерацию интерфейсов по порядку, то есть `vlan0`, `vlan1`, `vlan3` и тд);
- **create** - этой командой, собственно, и создается виртуальный интерфейс;
- **vlan 8** - указываем, какой VID будет у созданного интерфейса. В моем случае это 8;
- **vlandev nfe0** - указываем, на каком именно физическом интерфейсе будет создан виртуальный (в моем частном случае - это `nfe0`);



802.1Q VLAN - реализация 802.1Q VLAN на FreeBSD

Опубликовано muff.kiev.ua (<http://muff.kiev.ua>)

- **195.3.159.250 netmask 255.255.255.252** - присваиваем IP-адрес интерфейсу.

По аналогии создаем vlan12:

```
# ifconfig vlan12 create vlan 12 vlandev nfe0 10.25.25.1 netmask 255.255.255.0
```

Проверяем, что у нас получилось:

```
# ifconfig
```

```
nfe0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500      options=8<VLAN_MTU>      ether 00:22:15:02:04:8f      media: Ethernet autoselect (100baseTX <full-duplex>)      status: active      lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384      options=3<RXCSUM,TXCSUM>      inet 127.0.0.1 netmask 0xff000000      broadcast 195.3.159.251      media: Ethernet autoselect (100baseTX <full-duplex>)      status: active      vlan: 8 parent interface: nfe0      vlan12: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500      ether 00:22:15:02:04:8f      inet 195.3.159.250 netmask 0xffffffff      broadcast 195.3.159.251      media: Ethernet autoselect (100baseTX <full-duplex>)      status: active      vlan: 12 parent      interface: nfe0
```

Теперь дело за малым - внести необходимые изменения в rc.conf, чтобы не потерять настройки после перезагрузки сервера. Вот список необходимых опций:

```
ifconfig_nfe0="up" cloned_interfaces="vlan8 vlan12" ifconfig_vlan8="vlan 8 vlandev nfe0 195.3.159.250 netmask 255.255.255.252" ifconfig_vlan12="vlan 12 vlandev nfe0 10.25.25.1 netmask 255.255.255.0"
```

ifconfig_nfe0="up" - указываем, что интерфейс необходимо "поднять", иначе он не подымется самостоятельно. То есть виртуальные интерфейсы создаются, но трафик по ним проходить не будет.

cloned_interfaces="vlan8 vlan12" - необходимо перечислить, какие виртуальные интерфейсы необходимо создать.

Для удаления виртуального интерфейса необходимо воспользоваться ключем "destroy" команды ifconfig. Пример удаления vlan12:

```
# ifconfig vlan12 destroy
```

Cisco-like интерфейс

Исходные данные:

- коммутатор настроен корректно
- порт коммутатора в режиме "trunk"
- VID 7 - Internet
- VID 12 - локальная сеть
- сетевой адаптер на чипсете Realtek 8169 (re0)

Для поддержки 802.1Q VLAN на уровне ядра, необходимо наличие в конфигурационном файле ядра наличие следующей строки:



802.1Q VLAN - реализация 802.1Q VLAN на FreeBSD

Опубликовано muff.kiev.ua (<http://muff.kiev.ua>)

```
device      vlan
```

Если поддержки 802.1Q VLAN на уровне ядра нету, подгружим необходимый модуль ядра:

```
# kldload /boot/kernel/if_vlan.ko
# kldstat | grep vlan
```

```
4      1 0xffffffff80822000 2306      if_vlan.ko
```

Для автоматической подгрузки модуля во время загрузки операционной системы, необходимо внести изменения в **loader.conf**:

```
# echo 'if_vlan_load="YES"' >> /boot/loader.conf
```

Создадим vlan-интерфейс с VID 7:

```
# ifconfig re0.7 create 195.3.158.42 netmask 255.255.255.240
```

Более подробно о используемых опциях:

- **ifconfig** - команда конфигурирования сетевого интерфейса;
- **re0.7** - указываем "родительский" интерфейс (re0) и номер VID;
- **create** - этой командой, собственно, и создается виртуальный интерфейс;
- **195.3.158.42 netmask 255.255.255.240** - присваиваем IP-адрес интерфейсу.

По аналогии создадим интерфейс с VID 12:

```
# ifconfig re0.12 create 10.25.25.1 netmask 255.255.255.0
```

Проверяем, что у нас получилось в результате:

```
# ifconfig
```

```
re0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500          options=b<RX
CSUM,TXCSUM,VLAN_MTU>      ether 00:19:e0:61:e9:90      media: Ethernet autosele
ct (100baseTX <full-duplex>)      status: active lo0: flags=8049<UP,LOOPBACK,RUNNIN
G,MULTICAST> mtu 16384      inet 127.0.0.1 netmask 0xff000000 re0.7: flags=8843<UP,
BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500      inet 195.3.158.42 netmask 0xffff
ffff0 broadcast 195.3.158.47      ether 00:19:e0:61:e9:90      media: Ethernet a
utoselect (100baseTX <full-duplex>)      status: active      vlan: 7 parent inte
rface: re0 re0.12: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet 10.25.25.1 netmask 0xffffffff00 broadcast 10.25.25.255      ether 00:19:e0:61:
e9:90      media: Ethernet autoselect (100baseTX <full-duplex>)      status: act
ive      vlan: 12 parent interface: re0
```

Следующий шаг - внесение необходимых изменений в **rc.conf**, чтобы не потерять настройки после перезагрузки сервера:

```
cloned_interfaces="re0.7 re0.12"
ifconfig_re0="up"
```



802.1Q VLAN - реализация 802.1Q VLAN на FreeBSD

Опубликовано muff.kiev.ua (<http://muff.kiev.ua>)

```
ifconfig_re0_7="inet 195.3.158.42 netmask 255.255.255.240"
ifconfig_re0_12="inet 10.25.25.1 netmask 255.255.255.0"
```

ifconfig_re0="up" - указываем, что интерфейс необходимо "поднять", иначе он не подымется самостоятельно. То есть виртуальные интерфейсы создадутся, но трафик по ним проходить не будет.

cloned_interfaces="re0.7 re0.12" - необходимо перечислить, какие виртуальные интерфейсы необходимо создать.

Для удаления виртуального интерфейса необходимо воспользоваться ключем "**destroy**" команды **ifconfig**. Пример удаления `vlan12`:

```
# ifconfig re0.12 destroy
```

Источник (получено 2026-01-30 08:10):

<http://muff.kiev.ua/content/8021q-vlan-realizatsiya-8021q-vlan-na-freebsd>