

Настраиваем vlan на FreeBSD

VLAN (от англ. *Virtual Local Area Network*), VLAN могут являться частью большего LAN, имея определенные правила взаимодействия с другими VLAN, либо быть полностью изолированными от них. Простейший механизм изоляции различных подсетей, работающих через общие свичи и роутеры, известен как **802.1Q**.

Преимущества VLAN

- увеличивает число **широковещательных доменов**, но уменьшает размер каждого **широковещательного домена**, которые в свою очередь уменьшают сетевой трафик и увеличивают безопасность сети (оба следствия связаны вместе из-за единого большого широковещательного домена);
- уменьшает усилия администраторов на создание подсетей;
- уменьшает количество оборудования, так как сети могут быть разделены логически, а не физически;
- улучшает управление различными типами трафика.

Транк VLAN – это физический канал, по которому передается несколько VLAN каналов, которые различаются тегами (метками, добавляемыми в пакеты). Транки обычно создаются между «тегированными портами» VLAN-устройств: свитч-свитч или свитч-маршрутизатор. (В документах Cisco термином «транк» также называют объединение нескольких физических каналов в один логический: Link Aggregation, Port Trunking). Маршрутизатор (свитч третьего уровня) выступает в роли магистрального ядра сети (backbone) для сетевого трафика разных VLAN.

На устройствах Cisco, протокол VTP (VLAN Trunking Protocol) предусматривает VLAN-домены для упрощения администрирования. VTP также выполняет «чистку» трафика, направляя VLAN трафик

только на те коммутаторы, которые имеют целевые VLAN-порты.

Native VLAN – каждый порт имеет параметр, названный постоянный виртуальной идентификацией (Native VLAN), который определяет VLAN, назначенный получить нетеговые кадры.

Сказав проще, **vlan** – это логический канал внутри физического канала (кабеля), а **trunk** это множество логических каналов (vlan`ов) внутри одного физического канала (кабеля).

Итак более-менее с теорией разобрались, теперь подумаем зачем нам это может понадобиться.

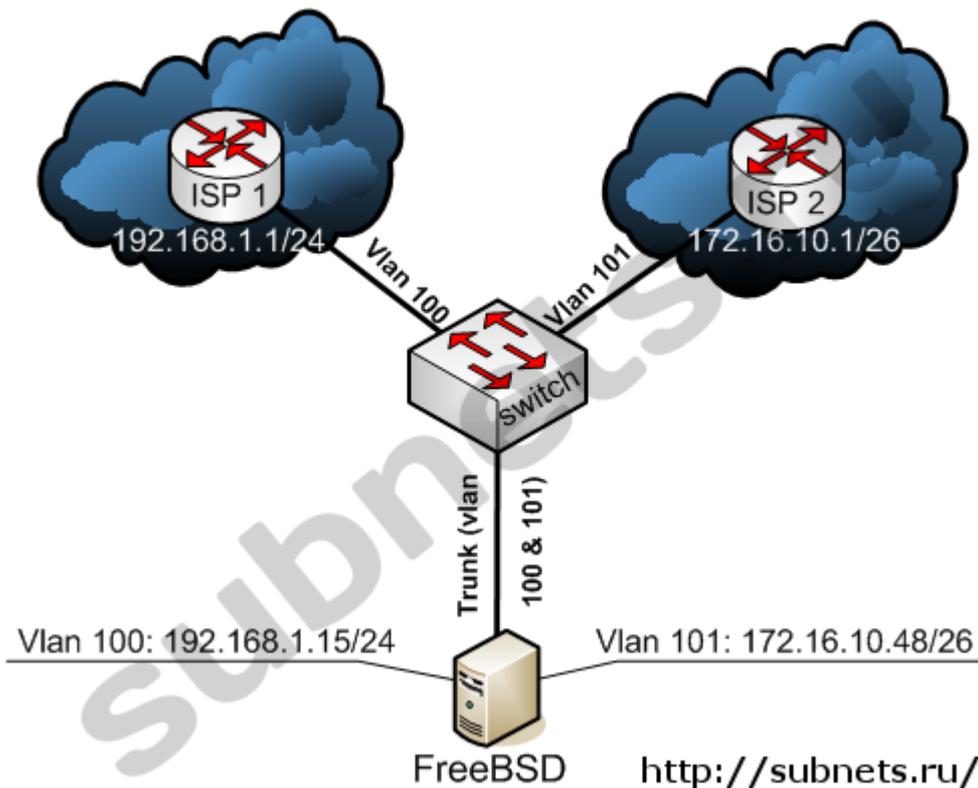
Данная технология может пригодиться например если на сервер нужно «подать» несколько физических линков, а сетевая карта всего одна и вставить ещё одну нет возможности.

Возьмем подобную ситуацию как пример и попробуем настроить следующее:

- У нас сервер только с одной сетевой картой, а необходимо подключить два канала от двух провайдеров
- Провайдер А выдал IP-адрес 192.168.1.15 маска 255.255.255.0
- Провайдер Б выдал IP-адрес 172.16.10.48 маска 255.255.255.192

Для того чтобы разрулить данную ситуацию нам понадобится switch который понимает Vlan (802.1Q), уже почти все управляемые свичи идут с этой функцией. В нашем примере рассмотрим два типа свичей:

1. cisco catalyst (например 2950 или 3560)
2. dlink DES-3526



Два провайдера и сервер FreeBSD с одной сет.картой

Начинаем

Воткнем физические связи в наш свич, получим три кабеля и три занятых порта

1. порт 1 – Провайдер А
2. порт 2 – Провайдер Б
3. порт 3 – наш сервер

Настроим cisco catalyst:

```

configure terminal
vlan 100
name provider_a
vlan 101
name provider_b
int gi0/1
switchport access vlan 100
int gi0/2
switchport access vlan 101
exit

```

```
exit
```

Этими командами мы создали два **vlan** с номерами 100 и 101 для линков от двух провайдеров и назначили два порта каталиста в эти **vlan**.

по команде `show vlan` вы должны видеть созданные **vlan`ы**

Теперь перейдем к конфигурированию 3-го порта каталиста куда воткнут наш сервер. Т.к. нам придется в этот порт посылать оба **vlan** (100,101) нам необходимо сделать **trunk** на этом порту:

```
configure terminal
int gi 0/3
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk allowed vlan 100,101
exit
exit
```

Этими командами мы на третьем порту каталиста подняли **trunk** и разрешили в этом **trunk`е** два **vlan** 100,101

В терминах Cisco:

- порт в аксес/аксес порт (access port) – порт принимающий **не** тегированные пакеты (пакеты в которых нет тега (номера) **vlan** которому они принадлежат)
- порт в транке/транк порт (trunk port) – порт принимающий тегированные пакеты в которых указан тег (номер) **vlan**

Сделаем тоже самое, но для Dlink:

```
create vlan provider_a tag 100
create vlan provider_b tag 101
config vlan provider_a add untagged 1
config vlan provider_b add untagged 2
config vlan provider_a add tagged 3
config vlan provider_b add tagged 3
```

Так же по команде `show vlan` убеждаемся что все на месте.

В терминах Dlink:

- антагет порт (untagged port) – порт в аксес режиме принимающий не тегированные пакеты
- тагет порт (tagged port) – порт в транке принимающий тегированные пакеты

Переходим к FreeBSD. В качестве примера используется сет. карта 82545EM Gigabit Ethernet Controller интерфейс em0

Для начала удалим все IP-адреса с интерфейса em0 (если они есть):

```
/sbin/ifconfig em0 delete
```

Создадим vlan для провайдера А:

```
/sbin/ifconfig vlan100 create  
/sbin/ifconfig vlan100 vlan 100 vlandev em0
```

Вот и все, **vlan** создан, проверяем есть ли он в списке интерфейсов:

запускаем команду `/sbin/ifconfig vlan100`

```
vlan100:      flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST>  
metric 0 mtu 1500  
options=3<RXCSUM,TXCSUM>  
ether 00:02:a5:4e:92:48  
media: Ethernet autoselect (100baseTX <full-duplex>)  
status: active  
vlan: 100 parent interface: em0
```

Итерфейс на месте.

Создадим **vlan** для провайдера Б:

```
/sbin/ifconfig vlan101 create  
/sbin/ifconfig vlan101 vlan 101 vlandev em0
```

Проверяем есть ли он в списке интерфейсов:

запускаем команду `/sbin/ifconfig vlan101`

```
vlan101: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST>  
metric 0 mtu 1500  
options=3<RXCSUM,TXCSUM>  
ether 00:02:a5:4e:92:48  
media: Ethernet autoselect (100baseTX <full-duplex>)  
status: active  
vlan: 101 parent interface: em0
```

Интерфейс на месте.

После того как интерфейсы **vlan`**ов созданы мы обращаемся с ними как с обычными интерфейсами обычных сетевых карт.

Добавим IP-адреса на созданные **vlan`**ы:

```
/sbin/ifconfig vlan100 add 192.168.1.15/24  
/sbin/ifconfig vlan101 add 172.16.10.48/26
```

Вот и все, если вы все сделали правильно, то при выводе команды `ifconfig` получите:

```
vlan100: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>  
metric 0 mtu 1500  
options=3<RXCSUM,TXCSUM>  
ether 00:02:a5:4e:92:48  
inet 192.168.1.15 netmask 0xffffffff00 broadcast 192.168.1.255  
media: Ethernet autoselect (100baseTX <full-duplex>)  
status: active  
vlan: 100 parent interface: em0
```

```
vlan101: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>  
metric 0 mtu 1500  
options=3<RXCSUM,TXCSUM>  
ether 00:02:a5:4e:92:48  
inet 172.16.10.48 netmask 0xffffffc0 broadcast 172.16.10.63  
media: Ethernet autoselect (100baseTX <full-duplex>)  
status: active  
vlan: 101 parent interface: em0
```

Можете проверять наличие связи с двумя провайдерами ?

Уничтожить/удалить `vlan` можно командой (например удалим `vlan100`):

```
/sbin/ifconfig vlan100 destroy
```

Осталось последнее дело, чтобы после reboot конфигурация vlan`ов сохранялась.

Для этого добавим в файл /etc/rc.conf следующие строчки:

```
ifconfig_vlan100="inet 192.168.1.15 netmask 255.255.255.0 vlan  
100 vlandev em0"  
ifconfig_vlan101="inet 172.16.10.48 netmask 255.255.255.192  
vlan 101 vlandev em0"  
cloned_interfaces="vlan100 vlan101"
```

Есть и второй способ сделать тоже самое. Создайте файл /etc/rc.local и в него вставьте все команды которые вы вводили для создания vlan`ов и присваивание им IP-адресов. Файл /etc/rc.local так же отработывается при загрузке сервера и будут исполнены все команды в нем перечисленные.

Из командной строки vlаны можно создавать так

```
ifconfig em0.100="inet 192.168.1.15 netmask 255.255.255.0"
```

[Источник](#)