LACP - обьединение сетевых интерфейсов с использованием Link Aggregation Control Protocol

Опубликовано muff в Пт, 2011-03-25 18:15

В связи с постоянным увеличением загруженности каналов, возникла необходимость в агрегировании портов между коммутирующим (ну и маршрутизирующим в том числе) оборудованием. Агрегирование каналов будем выполнять с поддержкой протокола **LACP**, который позволяет пропускать трафик, даже если один из физических линков выйдет из строя.

Примечание. Когда-то в построенном статическом **ether-channel** вышел из строя один из линков. Все было очень банально - вышел из строя один из медиаконвертеров. Ну и по закону упавшего бутерброда - на выходных, причем на стороне транспортного провайдера. Появились следующие баги - часть ресурсов не отвечает. Отловил только одну закономерность - пинговались только IP-адреса, в которых 4 байт был непарным. Долго думал, как это возможно. Но диагностировать проблему было очень трудно... Отловил ее только вызвонив админа транспортного провайдера, который и обнаружил, что с их стороны один из портов в статическом **ether-channel** "отвалился". Теперь стала понятной проблема с доступностью нечетных IP. Балансировка была настроена по алгоритму **dst-ip**, соответственно были недоступны хосты, доступ к которым шел через нерабочий канал. Все решилось заменой вышедшего из строя медиаконвертера и поглощением пива вместе с админом транспортного провайдера. Вот такая вот байка. А соль ее в том, что лучше все таки строить динамические агрегированные каналы, а не статические...

Итак, примеры построение агрегированного канала на разном оборудовании.

Примечание. ***Агрегирование каналов*** — технология, которая позволяет объединить несколько физических каналов в один логический. Такое объединение позволяет увеличивать пропускную способность и надежность канала. Агрегирование каналов может быть настроено как между двумя коммутаторами, так и между коммутатором и хостом.

FreeBSD

В FreeBSD агрегирование каналов возможно благодаря модулю ядра ***lagg***. Он поддерживает несколько режимов агрегирования:

- Failolver - работает один канал, если он упал, тогда второй
- Cisco FastEtherchannel
- LACP
- Round Robin

Как уже оговаривалось выше, будем использовать **LACP**.

Для поддержки lagg необходимо пересобрать ядро с такой строкой

```
device lagg
```

либо же добавить загрузку модуля в /boot/loader.conf:

```
if_lagg_load="YES"
```

Загрузить модуль не перезагружая сервер можно следующей командой:

```
# kldload /boot/kernel/if_lagg.ko
```

Проверяем, подгрузился ли модуль:

```
# kldstat

Id Refs Address            Size     Name
 1    2 0xffffffff80100000 5713f8   kernel
 2    1 0xffffffffb0717000 4916     if_lagg.ko
```

Модуль подгрузился. Теперь создадим агрегированный интерфейс из двух интерфейсов **em0** и **em1**:

```
# ifconfig em0 up
# ifconfig em1 up
# ifconfig lagg0 create laggproto lacp laggport em0 laggport em1 192.168.159.250
netmask 255.255.255.0
```

Для создания интерфейса во время загрузки ОС, добавим такие строки в rc.conf:

```
cloned_interfaces="lagg0"
ifconfig_em0="up"
ifconfig_em1="up"
ifconfig_lagg0="laggproto lacp laggport em0 laggport em1 192.168.159.250 netmask
255.255.255.0"
```

В результате получаем такой вот интерфейс:

```
# ifconfig lagg0

lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=19b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,TSO4>
        ether 00:15:17:5e:31:5c
        inet 192.168.159.250 netmask 0xffffff00 broadcast 192.168.192.255
        media: Ethernet autoselect
        status: active
        laggproto lacp
        laggport: em1 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
        laggport: em0 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
```

На этом настройку на стороне сервера можно считать оконченой.

Приведу еще пример построения lagg-интерфейса и использования vlan-ов (**802.1Q**). Для начала необходимо создать агрегированый интерфейс:

```
# ifconfig em0 up
# ifconfig em1 up
# ifconfig lagg0 create laggproto lacp laggport em0 laggport em1
```

Потом создаем виртуальные интерфейсы VLAN поверх агрегированого интерфейса с метками тегов 10 и 15:

```
# ifconfig vlan10 create vlan 10 vlandev lagg0 192.168.10.1 netmask 255.255.255.0
# ifconfig vlan15 create vlan 15 vlandev lagg0 192.168.15.1 netmask 255.255.255.0
```

В результате имеем такие интерфейсы:

```
# ifconfig vlan10
```

```
vlan10: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        ether 00:15:17:5e:31:5c
        inet 192.168.10.1 netmask 0xffffff00 broadcast 192.168.10.255
        media: Ethernet autoselect
        status: active
        vlan: 10 parent interface: lagg0

# ifconfig vlan15
vlan10: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        ether 00:15:17:5e:31:5c
        inet 192.168.15.1 netmask 0xffffff00 broadcast 192.168.15.255
        media: Ethernet autoselect
        status: active
        vlan: 15 parent interface: lagg0
```

Для того, чтобы все это "завелось" после перезагрузки сервера, в **rc.conf** необходимо добавить такие строки:

```
cloned_interfaces="lagg0 vlan10 vlan15"
ifconfig_em0="up"
ifconfig_em1="up"
ifconfig_lagg0="laggproto lacp laggport em0 laggport em1"
ifconfig_vlan10="vlan 10 vlandev lagg0 192.168.10.1 netmask 255.255.255.0"
ifconfig_vlan15="vlan 15 vlandev lagg0 192.168.15.1 netmask 255.255.255.0"
```

Примечание. В некоторых случаях при использовании 802.1Q агрегированный интерфейс не работал. "Лечится" это отключением опции **vlanhwtag** на сетевых интерфейсах. Тоесть, в таком случае команды "поднятия" интерфейсов **em0** и **em1** будут следующие:

```
# ifconfig em0 -vlanhwtag up
# ifconfig em1 -vlanhwtag up
```

И не забываем изменения внести в **rc.conf**:

```
ifconfig_em0="-vlanhwtag up"
ifconfig_em1="-vlanhwtag up"
```

Cisco Catalyst

Агрегирование портов будем выполнять на коммутаторе **Cisco Catalyst WS-C2960G-24TC-L**. Агрегировать будем порты **Gi0/1** и **Gi0/2**. Общие команды конфигурации наводить не буду. Наведу только команды, которые необходимо выполнить для агрегирования:

```
c2960(config)#port-channel load-balance dst-ip
c2960(config)#interface range GigabitEthernet 0/1-2
c2960(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1
c2960(config-if-range)#channel-protocol lacp
```

В результате этих команд мы установили режим балансировки, исходя из IP-адреса получателя (балансировка исходящего трафика в таком случае будет более равномерна) и создали интерфейс **Port-channel1**. Теперь все команды по конфигурированию необходимо применять к интерфейсу **Port-channel 1**. Эти изменения будут присвоены интерфейсам, которые входят в **channel-group 1** (в нашем частном случае, это **GigabitEthernet 0/1** и **GigabitEthernet 0/2**).

Для примера, переведем интерфейсы в режим транка и укажем, какие **VLAN** пропускать через этот транк:

```
c2960(config)#interface Port-channel 1
c2960(config-if)#switchport mode trunk
c2960(config-if)#switchport trunk allowed vlan 10,15
```

Примечание. Для того, чтобы поместить два интерфейсы в один и тот же **channel-group**, необходимо чтобы настройки конфигурации этих интерфейсов были идентичны.
Например, если порт **GigabitEthernet 0/1** будет в режиме **access**, а **GigabitEthernet 0/2** в режиме **trunk**, то попытка агрегировать эти интерфейсы закончится неудачей.

Alcatel Lucent

OS6850-24X

Агрегирование портов будем выполнять на коммутаторе **Alcatel Lucent OS6850-24X**. Агрегировать будем порты **Gi1/1** и **Gi1/2**. Команды для создания агрегированого канала следующие:

```
-> lacp linkagg 1 size 2 admin state enable
-> lacp linkagg 1 name "LACP 1"
-> lacp linkagg 1 actor admin key 1
-> lacp agg 1/1 actor admin key 1
-> lacp agg 1/2 actor admin key 1
```

В результате имеем интерфейс **linkagg 1**.

OmniStack LS 6224

Агрегирование портов будем выполнять на коммутаторе **Alcatel OmniStack LS 6224**. Агрегировать будем порты **g1** и **g2**. Команды для создания агрегированого канала следующие:

```
alc6224(config)# interface range ethernet g1-2
alc6224(config-if)# channel-group 1 mode auto
```

Foxgate

Агрегирование портов будем выполнять на коммутаторе **Foxgate S9816-GS8-X2**. Агрегировать будем порты **Eth1/1** и **Eth1/2**. Команды для создания агрегированого канала следующие:

```
9816-sw(config)#port-group 1
9816-sw(config)#interface ethernet 1/1-2
9816-sw(config-if-port-range)#port-group 1 mode active
9816-sw(config-if-port-range)#interface port-channel 1
9816-sw(config-if-port-channel1)#name LACP1
9816-sw(config-if-port-channel1)#load-balance dst-ip
```

В результате имеем в своем распоряжении интерфейс **Port-Channel1**.

D-Link

Агрегирование портов будем выполнять на коммутаторе **D-Link DES-3828**. Агрегировать будем порты 1 и 2. Команды для создания агрегированого канала следующие:

```
DES-3800:admin#config link_aggregation algorithm ip_destination
DES-3800:admin#create link_aggregation group_id 1 type lacp
DES-3800:admin#config link_aggregation group_id 1 master_port 1 ports 1-2 state enable
DES-3800:admin#config lacp_port 1-2 mode active
```

Juniper

Агрегирование портов будем выполнять на маршрутизаторе **Juniper MX80** под управлением ОС **Junos**. Агрегировать будем порты **ge-1/0/1** и **ge-1/0/2**.

```
root@core# set chassis aggregated-devices ethernet device-count 1

[edit]

root@core# set interfaces ge-1/0/1 gigether-options 802.3ad ae0

[edit]

root@core# set interfaces ge-1/0/2 gigether-options 802.3ad ae0

[edit]

root@core# set interfaces ae0 aggregated-ether-options lacp active

[edit]

root@core# commit
commit complete
```

Edge-Core

Особенности настройки агрегации каналов в коммутаторах Edge-Core:

- перед созданием физического подключения между портами двух коммутаторов необходимо завершить все необходимые настройки **LACP** для предотвращения образование петель;
- на коммутаторах **E3552M** можно создать до 8 виртуальных транков, а каждый транк может включать в себя до 8 портов;
- порты на двух концах виртуального канала должны быть настроены в режиме транк;
- порты на двух концах транка должны быть сконфигурированы одинаково, включая режимы связи (например, скорость, дуплексный режим или flow control), **VLAN** и **QoS**;
- каждый из гигабитных портов может быть включен в один виртуальный канал с другими гигабитными портами, не зависимо от того, какой из Combo-портов используется (SFP или медный);
- **STP**, **VLAN** и настройки **IGMP** могут быть применены только ко всем портам в транке.

Агрегирование портов будем выполнять на коммутаторе **Edge-Core ES3552M**. Агрегировать будем порты **ethernet 1/32** и **ethernet 1/33**.

```
Vty-0(config)#interface ethernet 1/32-33
Vty-0(config-if)#lacp
```

Механизм распределения сетевого трафика по виртуальным каналам транка: выбор, по какому физическому каналу в виртуальном транке будет передаваться сетевой трафик, осуществляется с помощью хеш-функции на базе следующих параметров: **MAC** адреса

получателя (**MACSA**), **MAC** адреса отправителя (**MACDA**), **VLAN**, типа пакета **Ethernet** и **IP** адреса отправителя (**IPSA**). Механизм расчета основывается на формуле, которая возвращает 3-хбитный индекс, указывающий на физический канал, по которому будет передан пакет.

Eltex

Агрегирование портов будем выполнять на коммутаторе **Eltex MES2324FB**. Агрегировать будем порты 20 и 21. Команды для создания агрегированого канала следующие:

```
# KL-L9-SW1(config)# port-Channel load-balance src-dst-mac-ip

# KL-L9-SW1(config)# interface range GigabitEthernet 1/0/20 - 21

# KL-L9-SW1(config-if-range)# channel-group 1 mode auto
```

Источник (получено *2026-11-26 18:33*):
http://muff.kiev.ua/content/lacp-obedinenie-setevykh-interfeisov-s-ispolzovaniem-link-aggregation-control-protocol