

Автоматическое переключение каналов

Ведение

Рано или поздно практически каждый системный администратор сталкивается с задачей использования 2х и более каналов и автоматического переключения маршрутизации при падении одного из них на резервный канал. К сожалению на linux нет т.н. продакшен решения для решения этой задачи. Единственное решения - это использование протоколов динамической маршрутизации таких как BGP, который в свою очередь требует наличия AS. Что к сожалению не всегда возможно чисто технически и экономически. Поэтому и приходится обходить эти моменты по другому.

Итак рассмотрим классический случай. Есть шлюз на базе GNU Linux, к которому одновременно подключены 2 интернет провайдера. Наша задача - при падении основного шлюза, переключать автоматически всех на резервный канал, а после восстановления делать обратную операцию.

Для этого будем использовать небольшой скрипт. Принцип его очень простой. Раз в n минут мы проверяем доступность шлюза провайдера с помощью утилиты ping. И если основной шлюз не доступен или процент потерь пакетов больше заданного порога, то мы меняем шлюз по умолчанию на резервный. После переключения на резервный канал мы также продолжаем проверять доступность основного шлюза и после того, как он снова станет доступным снова переключаемся на него.

Написание скрипта

Ниже привожу сам скрипт

```
#!/bin/sh

# Указываем путь к log-файлу
log="/var/log/routing.log"

# Задаем периодичность опроса шлюзов.
# 30s - 30 секунд
# 10m - 10 минут
# 1h - 1 час
check_period="30s"

# Если log-файл не найден, создаем его, иначе отчищаем
clear_log ()
{
if [ ! -f ${log} ]
then
touch ${log}
```

```
else
    cat /dev/null > ${log}
fi
}

# Инициализируем переменные
init ()
{
    gw1="192.168.1.1"
    gw2="192.168.2.1"
    preferred_gw=${gw1}

    dev_if1="eth1"
    dev_if2="eth2"

    if1=`ifconfig ${dev_if1} | awk -F ' *|:' '/inet addr/{print $4}'`
    if2=`ifconfig ${dev_if2} | awk -F ' *|:' '/inet addr/{print $4}'`

# Текущие статусы шлюзов: 0 - шлюз недоступен, 1 - шлюз работает
    gw1_curr_status=0
    gw2_curr_status=0

# Текущие потери пакетов на шлюзе
    gw1_curr_packet_loss=0
    gw2_curr_packet_loss=0

# Максимальный процент потерь пакетов, при превышении данного числа, шлюз считается
недоступным
    gw1_max_packet_loss=35
    gw2_max_packet_loss=40

# Отчищаем log-файл и сохраняем значения основных переменных
    clear_log
    echo `date +"%T %d.%m.%Y"`. " Init environment OK. Check gateways status
every ${check_period}." >> ${log}
    echo `date +"%T %d.%m.%Y"`. " ISP1 [if1=${if1}, dev_if1=${dev_if1},
gw1=${gw1}], ISP2 [if2=${if2}, dev_if2=${dev_if2}, gw2=${gw2}]" >> ${log}
}

# Данная функция определяет текущее состояние каждого из шлюзов.
# Если текущий процент потерь меньше максимально допустимого, то считаем, что шлюз
доступен.
get_current_status ()
{
    echo `date +"%T %d.%m.%Y"`. " Get current status ISP gateways." >> ${log}

# Get current status default gateway ISP1
    gw1_curr_packet_loss=`ping -I ${dev_if1} -c20 -l20 -q -W3 ${gw1} | grep
loss | awk '{print $(NF-4)}' | cut -d"%" -f1`
    if [ ${gw1_curr_packet_loss} -le ${gw1_max_packet_loss} ]
    then
```

```

        echo `date +"%T %d.%m.%Y"` . "ISP1. [STATUS - OK]. Current packet
loss on ${gw1} via ${dev_if1} is ${gw1_curr_packet_loss}%" >> ${log}
        gw1_curr_status=1
    else
        echo `date +"%T %d.%m.%Y"` . "ISP1. [STATUS - CRITICAL]. Current
packet loss on ${gw1} via ${dev_if1} is ${gw1_curr_packet_loss}%" >> ${log}
        gw1_curr_status=0
    fi

# Get current status default gateway ISP2
gw2_curr_packet_loss=`ping -I ${dev_if2} -c20 -l20 -q -W3 ${gw2} | grep
loss | awk '{print $(NF-4)}' | cut -d"%" -f1`
if [ ${gw2_curr_packet_loss} -le ${gw2_max_packet_loss} ]
then
    echo `date +"%T %d.%m.%Y"` . "ISP2. [STATUS - OK]. Current packet
loss on ${gw2} via ${dev_if2} is ${gw2_curr_packet_loss}%" >> ${log}
    gw2_curr_status=1
else
    echo `date +"%T %d.%m.%Y"` . "ISP2. [STATUS - CRITICAL]. Current
packet loss on ${gw2} via ${dev_if2} is ${gw2_curr_packet_loss}%" >> ${log}
    gw2_curr_status=0
fi
}

# Данная функция производит переключение шлюза по умолчанию в зависимости от
результатов полученных в get_current_status ()
# На данный момент предпочитаемым шлюзом может быть только ${gw1}
switch_default_gw ()
{
    curr_gw=`ip route show | grep default | awk '{print $3}'`
    if [ ${curr_gw} = ${prefered_gw} -a ${gw1_curr_status} -eq 1 ]
    then
        echo `date +"%T %d.%m.%Y"` . "[STATUS - OK]. Current default gateway
is prefered and running up. Nothing to do." >> ${log}
        return
    fi

    if [ ${curr_gw} = ${prefered_gw} -a ${gw1_curr_status} -eq 0 -a
${gw2_curr_status} -eq 1 ]
    then
        echo `date +"%T %d.%m.%Y"` . "Prefered gw ${gw1} is down. Change
default gw to ${gw2}" >> ${log}
        echo `date +"%T %d.%m.%Y"` . "ip route replace default via ${gw2} dev
${dev_if2}" >> ${log}
        /sbin/ip route replace default via ${gw2} dev ${dev_if2}
        return
    fi

    if [ ${curr_gw} != ${prefered_gw} -a ${gw1_curr_status} -eq 1 ]
    then
        echo `date +"%T %d.%m.%Y"` . "Prefered gw ${prefered_gw} is running

```

```

up now. Change default gw ${curr_gw} to preferred ${preferred_gw}" >> ${log}
    echo `date +"%T %d.%m.%Y"` . "ip route replace default via ${gw1} dev
${dev_if1}" >> ${log}
    /sbin/ip route replace default via ${gw1} dev ${dev_if1}
    return
fi

    if [ ${curr_gw} != ${preferred_gw} -a ${gw1_curr_status} -eq 0 -a
${gw2_curr_status} -eq 1 ]
    then
        echo `date +"%T %d.%m.%Y"` . "Preferred gw ${preferred_gw} is still
down. Current default gw is ${gw2}" >> ${log}
        return
    fi

    if [ ${gw1_curr_status} -eq 0 -a ${gw2_curr_status} -eq 0 ]
    then
        echo
"*****"
>> ${log}
        echo `date +"%T %d.%m.%Y"` . "CRITICAL. Two gateways is down. Try
again later." >> ${log}
        echo
"*****"
>> ${log}
        return
    fi
}

# Инициализируем переменные
init

# "Заходим" в вечный цикл, в котором по очереди вызываем функции get_current_status
() и switch_default_gw ()
# После этого делаем паузу, время паузы задается в переменной ${check_period}
while [ 1 ]
do
    get_current_status
    switch_default_gw
    sleep ${check_period}
done

```

Для запуска самого скрипта достаточно выполнить следующую команду

```
# ./check_gateways.sh &
```

Для автоматического запуска скрипта во время загрузки добавляем следующую строку в **/etc/rc.d/rc.local**

```
# cat /etc/rc.d/rc.local
```

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
/usr/local/bin/check_gateways.sh &
```

Для того, чтобы данная реализация работало вам необходимо будет использовать MASQUERADE без указания выходного интерфейса вместо SNAT.

```
# iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j MASQUERADE
```

Тестирование

Два шлюза доступны, нормальная ситуация

```
16:24:18 11.10.2009. Init environment OK. Check gateways status every 30s.
16:24:18 11.10.2009. ISP1 [if1=192.168.1.2, dev_if1=eth0, gw1=192.168.1.1],
ISP2 [if2=192.168.2.2, dev_if2=eth1, gw2=192.168.2.1]
16:24:18 11.10.2009. Get current status ISP gateways.
16:24:18 11.10.2009. ISP1. [STATUS - OK]. Current packet loss on 192.168.1.1
via eth0 is 0%.
16:24:18 11.10.2009. ISP2. [STATUS - OK]. Current packet loss on 192.168.2.1
via eth1 is 0%.
16:24:18 11.10.2009. [STATUS - OK]. Current default gateway is preferred and
running up. Nothing to do.
```

Шлюз второго провайдера недоступен, но как видно, переключение не происходит, так как его шлюз не является предпочитаемым

```
16:24:48 11.10.2009. Get current status ISP gateways.
16:24:48 11.10.2009. ISP1. [STATUS - OK]. Current packet loss on 192.168.1.1
via eth0 is 0%.
16:24:51 11.10.2009. ISP2. [STATUS - CRITICAL]. Current packet loss on
192.168.2.1 via eth1 is 100%.
16:24:51 11.10.2009. [STATUS - OK]. Current default gateway is preferred and
running up. Nothing to do.
```

Шлюз второго провайдера снова доступен

```
16:25:21 11.10.2009. Get current status ISP gateways.
16:25:21 11.10.2009. ISP1. [STATUS - OK]. Current packet loss on 192.168.1.1
via eth0 is 0%.
16:25:21 11.10.2009. ISP2. [STATUS - OK]. Current packet loss on 192.168.2.1
via eth1 is 0%.
16:25:21 11.10.2009. [STATUS - OK]. Current default gateway is preferred and
running up. Nothing to do.
```

Предпочитаемый шлюз недоступен, происходит переключение на второй

```
16:25:51 11.10.2009. Get current status ISP gateways.  
16:25:54 11.10.2009. ISP1. [STATUS - CRITICAL]. Current packet loss on  
192.168.1.1 via eth0 is 100%.  
16:25:54 11.10.2009. ISP2. [STATUS - OK]. Current packet loss on 192.168.2.1  
via eth1 is 0%.  
16:25:54 11.10.2009. Prefered gw 192.168.1.1 is down. Change default gw to  
192.168.2.1  
16:25:54 11.10.2009. ip route replace default via 192.168.2.1 dev eth1
```

Предпочитаемый шлюз снова доступен, переключаемся на него

```
16:26:24 11.10.2009. Get current status ISP gateways.  
16:26:24 11.10.2009. ISP1. [STATUS - OK]. Current packet loss on 192.168.1.1  
via eth0 is 0%.  
16:26:24 11.10.2009. ISP2. [STATUS - OK]. Current packet loss on 192.168.2.1  
via eth1 is 0%.  
16:26:24 11.10.2009. Prefered gw 192.168.1.1 is running up now. Change  
default gw 192.168.2.1 to prefered 192.168.1.1  
16:26:24 11.10.2009. ip route replace default via 192.168.1.1 dev eth0
```

Недоступны два шлюза. Тут мы уже ничего не поделаем, только записываем сообщение в log-файл. В принципе можно отправлять sms, если у вас есть такая возможность, так как ситуация критическая.

```
16:26:54 11.10.2009. Get current status ISP gateways.  
16:26:57 11.10.2009. ISP1. [STATUS - CRITICAL]. Current packet loss on  
192.168.1.1 via eth0 is 100%.  
16:27:00 11.10.2009. ISP2. [STATUS - CRITICAL]. Current packet loss on  
192.168.2.1 via eth1 is 100%.  
*****  
16:27:00 11.10.2009. CRITICAL. Two gateways is down. Try again later.  
*****
```

Второй шлюз снова доступен, при этом предпочитаемый пока что нет, переключаемся на доступный шлюз

```
16:27:30 11.10.2009. Get current status ISP gateways.  
16:27:33 11.10.2009. ISP1. [STATUS - CRITICAL]. Current packet loss on  
192.168.1.1 via eth0 is 100%.  
16:27:33 11.10.2009. ISP2. [STATUS - OK]. Current packet loss on 192.168.2.1  
via eth1 is 0%.  
16:27:33 11.10.2009. Prefered gw 192.168.1.1 is down. Change default gw to  
192.168.2.1  
16:27:33 11.10.2009. ip route replace default via 192.168.2.1 dev eth1
```

А теперь у нас снова доступен и предпочитаемый шлюз, делаем переключение на него

```
16:28:03 11.10.2009. Get current status ISP gateways.  
16:28:03 11.10.2009. ISP1. [STATUS - OK]. Current packet loss on 192.168.1.1  
via eth0 is 0%.  
16:28:03 11.10.2009. ISP2. [STATUS - OK]. Current packet loss on 192.168.2.1  
via eth1 is 0%.  
16:28:03 11.10.2009. Prefered gw 192.168.1.1 is running up now. Change  
default gw 192.168.2.1 to prefered 192.168.1.1  
16:28:03 11.10.2009. ip route replace default via 192.168.1.1 dev eth0
```

Примечание

Данный скрипт не будет работать на FreeBSD, так как используется утилита ip. Но вам никто не мешает и не запрещает доработать скрипт и заменить вызов ip на команды, которые доступны в вашем дистрибутиве.

Полезные ссылки

- <http://habrahabr.ru/post/267339/>

From:
<http://sys-adm.org.ua/> - **wiki.sys-adm.org.ua**

Permanent link:
<http://sys-adm.org.ua/net/routing>

Last update: **2017/08/02 15:12**

