Load Balancing - Single Multipath Route HOWTO

Shakthi Kannan, shaks_wants_no_spam_at_shakthimaan_dot_com

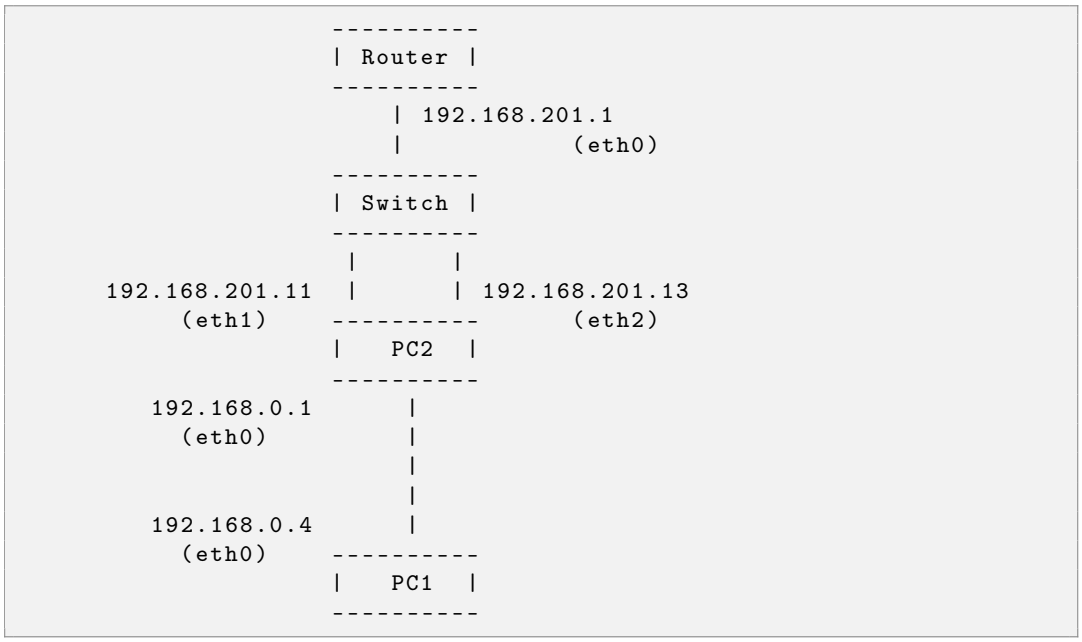
January 5, 2007

Revision: 1.2

Abstract

This documentation provides the steps to setup load-balancing for two NIC cards that are connected to a common router. The Redhat 9.0 (shrike) distribution is used for testing on x86 hardware.

1 Network Diagram



1.1 Install the kernel sources

Redhat 9.0 is used for testing, and the kernel sources are not installed by default. Redhat 9.0 uses the 2.4.20-8 kernel. You can get the:

```
kernel-source-2.4.20-8.i386.rpm
```

from the 2nd installation CD and install it using:

```
rpm -ivh kernel-source-2.4.20-8.i386.rpm
```

The kernel sources will get installed in /usr/src/linux-2.4.20-8.

1.2 Build iptables

The iptables version present in Redhat 9.0 does not have the –to-source option for SNAT. Hence, you need to download iptables v1.2.11. Extract the same and run the following in its sources directory:

```
make KERNEL_DIR=/usr/src/linux-2.4.20-8
make install KERNEL_DIR=/usr/src/linux-2.4.20-8
```

1.3 Run patch-o-matic

patch-o-matic-ng does not work with 2.4.20 Linux kernels. Hence, you need to download patch-o-matic-20031219.tar.bz2. Extract the same and do:

```
KERNEL_DIR=/usr/src/linux-2.4.20-8 \
  IPTABLES_DIR=/path/to/iptables/sources ./runme base
```

It will prompt for applying various patches. Say N to all, except for the random.patch.

1.4 Recompile the kernel

The 2.4.20-8custom kernel will be built when you recompile the kernel.

```
make clean
make dep
make bzImage
make modules
make modules_install
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.20-8custom
mkinitrd /boot/initrd-2.4.20-8custom.img 2.4.20-8custom
```

1.5 Update grub

Update the grub boot-loader, /boot/grub/menu.lst file:

```
title Redhat GNU/Linux (2.4.20-8custom)
       root (hd0,0)
       kernel /vmlinuz-2.4.20-8custom ro root=LABEL=/
       initrd /initrd-2.4.20-8custom.img
```

Reboot into the new kernel.

1.6 Configure ethernet

On PC2:

```
ifconfig eth0 192.168.0.1
ifconfig eth2 192.168.201.13
ifconfig eth1 192.168.201.11
```

On PC1:

```
ifconfig eth0 192.168.0.4
```

1.7 Load the driver modules

Check if the network driver modules are loaded, else load them using modprobe or insmod. Load the ip_tables.o and ipt_random.o drivers.

1.8 iptables script

Create the loadbalance.sh script.

```
# Iptables userspace executable
IPTABLES="/path/to/iptables-sources/iptables"
# Internal Interface
NET_INT_INT=eth0
# Internal IP
NET_INT_IP=192.168.0.1
# Internal Subnet
NET_INT_SUB=/24
# Internal Network
NET_INT_NET=192.168.0.0
# First external interface
NET_EXT_INT1=eth1
# First external IP
NET_EXT_IP1=192.168.201.11
# First external interface's gateway
NET_EXT_GW1=192.168.201.1
# Second external interface
NET_EXT_INT2=eth2
# Second external IP
NET_EXT_IP2=192.168.201.13
# Second external interface's gateway
NET_EXT_GW2=192.168.201.1
echo "Flushing All Tables"
$IPTABLES -F
$IPTABLES -F -t nat
$IPTABLES -F -t mangle
$IPTABLES -X -t nat
$IPTABLES -X -t mangle
$IPTABLES -X

$IPTABLES -t mangle -N ETH1
$IPTABLES -t mangle -F ETH1

$IPTABLES -t mangle -A ETH1 -p tcp -j LOG --log-prefix \
         "MANGLE_TCP_ETH1 "

$IPTABLES -t mangle -A ETH1 -p icmp -j LOG --log-prefix \
         "MANGLE_ICMP_ETH1 "

$IPTABLES -t mangle -A ETH1 -j MARK --set-mark 1

$IPTABLES -t mangle -N ETH2
$IPTABLES -t mangle -F ETH2
```

```
$IPTABLES -t mangle -A ETH2 -p tcp -j LOG --log-prefix \
        "MANGLE_TCP_ETH2 "

$IPTABLES -t mangle -A ETH2 -p icmp -j LOG --log-prefix \
        " MANGLE_ICMP_ETH2 "

$IPTABLES -t mangle -A ETH2 -j MARK --set-mark 2

$IPTABLES -t nat -N SPOOF_ETH1
$IPTABLES -t nat -F SPOOF_ETH1

$IPTABLES -t nat -A SPOOF_ETH1 -j LOG --log-prefix \
        "SPOOF_ETH1 "

$IPTABLES -t nat -A SPOOF_ETH1 -j SNAT --to-source \
        192.168.201.11

$IPTABLES -t nat -N SPOOF_ETH2
$IPTABLES -t nat -F SPOOF_ETH2

$IPTABLES -t nat -A SPOOF_ETH2 -j LOG --log-prefix \
        "SPOOF_ETH2 "

$IPTABLES -t nat -A SPOOF_ETH2 -j SNAT --to-source \
        192.168.201.13

echo "Setting some local network rules..."

$IPTABLES -A INPUT -p icmp -s 192.168.0.0/24 -d 192.168.0.1 \
        -j ACCEPT

echo "Setting Mangle rules for eth1..."

$IPTABLES -t mangle -A OUTPUT -o ! eth0 -m random --average 50 \
        -j ETH1

$IPTABLES -t mangle -A PREROUTING -i eth0 -m random --average 50 \
        -j ETH1

ip ro add table 10 default via 192.168.201.1 dev eth1
ip ru add fwmark 1 table 10
ip ro fl ca

echo "Setting Mangle rules for eth2..."

$IPTABLES -t mangle -A OUTPUT -o ! eth0 -m random --average 50 \
        -j ETH2

$IPTABLES -t mangle -A PREROUTING -i eth0 -m random --average 50 \
        -j ETH2

ip ro add table 20 default via 192.168.201.1 dev eth2
ip ru add fwmark 2 table 20
```

```
ip ro fl ca

echo "Setting up spoofing rules..."
$IPTABLES -t nat -A POSTROUTING -o eth1 -j SPOOF_ETH1
$IPTABLES -t nat -A POSTROUTING -o eth2 -j SPOOF_ETH2

echo "Adding default route..."
ip ro add default nexthop via 192.168.201.1 dev eth1 weight 1 \
         nexthop via 192.168.201.1 dev eth2 weight 1

echo "Disabling Reverse Path Filtering..."
echo 0> /proc/sys/net/ipv4/conf/eth1/rp_filter
echo 0> /proc/sys/net/ipv4/conf/eth2/rp_filter

echo "Enabling IPv4 Packet forwarding..."
echo "1"> /proc/sys/net/ipv4/ip_forward
```

Run it.

```
sh loadbalance.sh
```

1.9 Testing

Add a default route for PC2 from PC1:

```
route add default gw 192.168.0.1 eth0
```

Ping the gateway IP address from PC1:

```
ping 192.168.201.1
```

You should see packets going out from eth1 and eth2.

2 Bibliography

Hisham . Hisham Mardam Bey. July 26, 2004. Load Balancing across Multiple Links. http://www.linux.com.lb/wiki/index.pl?node=HOW-TOs.

Chris . Chris Lowth. April 1, 2004. The Hidden Treasures of iptables. http://www.linuxjournal.com/article/7180.