Balanceamento de links

Utilizando iptables e iproute2

Leandro R.
bolinh0@click21.com.br

Introdução:

Com o barateamento de links de acesso a internet e o aparecimento da conexão de banda-larga a um valor acessível, muitas empresas, e edifícios residenciais, começaram a utilizar soluções para prover o acessos a internet através de pequenos servidores (routers) que faziam a comunica ção entre a rede do cliente com a internet. Nesse momento surgiu a idéia de se juntar duas conexões com a internet de modo que um unico canal de saída fosse visto, fazendo com que a velocidade da conexão fosse duplicada, consequentemente barateando o custo de uma conexão de por exemplo 1Mbps, saindo pelo valor de 2 conexões de ADSL de 512Kbps.

Para por em pratica a idéia de se utilizar o balancemento de conexão com a internet, e até mesmo o serviço de redundancia a falhas contra queda de conexão foi elaborado este "mini HOW-TO" que demonstra de forma simples e clara de como este tipo de configuração pode ser feita, e dar o chute inicial para que outros scripts com mais funcionalidades e mais complexos possam ser criados.

Requisitos:

Alguns requisitos são vitais para o funcionamento deste tipo de configuração tais como os ja descritos no título do artigo:

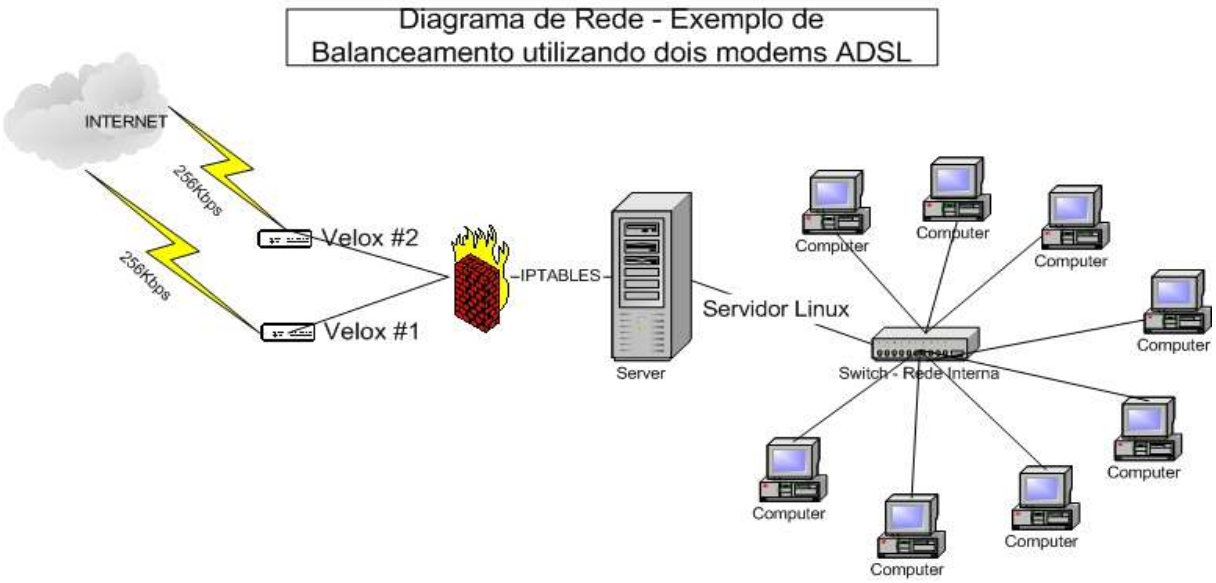
- Iptables
- Iproute2
- Scripts de inicialização das conexões (caso seja velox, ou serviço parecido)

Cenário:

O cenário utilizado para a demonstração do exemplo, é o de um edifício residencial, no qual existem duas conexões com a internet através de assinaturas de acesso a banda-larga (Velox), os dois modems ADSL estão configurados como ROTEADORES, existe um servidor linux (distribuição a escolha) com 3 placas de rede, que estão configuradas da seguinte maneira:

eth0: Rede interna (Rede do edifício) - (192.168.0.1/255.255.255.0)
eth1: Conexão Velox #1 - (192.168.1.2/255.255.255.0)
eth2: Conexão Velox #2 - (192.168.2.2/255.255.255.0)

O dispositivo de rede eth0 está ligado a um switch, onde as demais máquinas dos apartamentos estão ligadas.



OBS: Por este diagrama é visível a configuração de que todos os computadores da rede interna poderão se "enxergar", pois estão na mesma rede.

Configurações:

Primeiro Passo: Inicializando as conexões.

Neste ponto começarei a mostrar os scripts utilizados para a configuração do servidor. Vou começar mostrando algumas configurações que foram feitas dentro do /etc/rc.local, que acabaram sendo de grande ajuda para que o serviço funcionasse corretamente.

---------- /etc/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

echo "Inicializando conexao com a internet.."
/sbin/ifdown eth1
/sbin/ifdown eth2

echo "Iniclizando VELOX #1"
/sbin/ifup eth1
/sbin/route add default gw 192.168.1.1
/etc/firewall/ip-velox.pl

echo "Inicializando VELOX #2"
/sbin/ifup eth2
/sbin/route del default gw 192.168.1.1
/sbin/route add default gw 192.168.2.1
/etc/firewall/ip-velox.pl

/sbin/route del default gw 192.168.2.1

echo "Configurando o firewall"
/etc/firewall/firewall.sh
```

---------- /etc/rc.local ----------

Essas configurações foram feitas para que o velox pudesse se conectar automaticamente, usando o script ip-velox.pl (Desenvolvido pelo Fabio Vilan, e pode ser encontrado em http://www.isec.com.br/velox)

OBS: Não sei informar se este script ainda consegue se conectar ao velox, o legal dele, é a parte de teste de conexão (testa se está ativa ou não).

As configurações das rotas foram necessarias para que cada velox pudesse se conectar e autenticar, pois o problema que geralmente acontece quando se tem duas conexões no mesmo computador, é de que, quando a primeira conexão é estabelecida, a rota padrão é gerada, assim a segunda conexão fica perdida, tentando enviar pacotes que usam a rota padrão, no caso, pela primeira que já está estabelecida, assim não conseguiamos autenticar a segunda conexão do velox.

Lendo-se o script é possível identificar os passos para o estabelecimento das conexões:

1) Derruba as duas conexões (uma de cada velox);
2) Levanta a placa do primeiro velox (Velox #1);
3) Adiciona uma rota padrão para esta placa que foi inicializada;
4) Tenta se conectar e autenticar utilizando o script ip-velox.pl
5) Levanta a placa do segundo velox (Velox #2);
6) Deleta a rota padrão criada para o primeiro velox (Velox #1);
7) Adiciona a rota padrão para o segundo velox (Velox #2);
8) Utiliza o script de conexão e autenticação para o segundo velox (Velox #2);
9) Deleta a rota padrão criada para o segundo velox (Velox #2);
10) Levanta o firewall.

OBS: Este script termina por não estabelecer uma rota padrão, pois estas configurações devem ser feitas dentro do script que é chamado pelo rc.local, no caso o /etc/firewall/firewall.sh.

Segundo Passo: Criando tabelas de conexões para roteamento.

Quando o iproute2 está instalado, é criado um arquivo dentro do diretório /etc/iproute2 (ou /etc), chamado rt_tables. O arquivo onde as tabelas (de regras) de roteamento são definidas.

Cada tabela é definida por seu número identificador e nome. A ordenação vai de 0 à 255 (256 valores = 8 bits) e a faixa de 253 à 255 é reservada às tabelas do sistema (local, main e default).

O kernel trabalha exclusivamente com o identificador númerico da tabela. Assim Podemos estabelecer novas tabelas e definir situações especiais de roteamento.

Exemplo de arquivo (/etc/iproute2/rt_tables) utilizado para a configuração do balanceamento de links de acordo com o cenário proposto.

---------- /etc/iproute2/rt_tables ----------

```
#
# reserved values
#
#255     local
#254     main
#253     default
#0       unspec

#
# local
#
#1       inr.ruhep
10       velox1
11       velox2
30       velox
```

---------- /etc/iproute2/rt_tables ----------

Neste exemplo foram criadas 3 tabelas que serão utilizadas para a criação das regras de roteamento avançadas, necessárias para o funcionamento do balanceamento de links.

As tabelas cridas no arquivo rt_tables foram:

Valor	Nome
10	velox1
11	velox2
30	velox

De acordo com o arquivo, uma tabela tem “peso” 10, outra tem “peso” 11 e outra tabela, com valor mais acima tem “peso” 30. A criação dessas tabelas foram necessárias, pois são utilizadas no momento em que serão criadas as regras de roteamento.

Para maiores explicações sobre rt_tables, e roteamento avançado, procure pelos documentos da RNP (GTER), http://eng.registro.br/gter17/videos/05-roteamento-avancado-linux.pdf, em sites de busca (google) www.google.com..br, tambem vale a pena dar uma olhada no how-to do iproute2, além disso existe uma empresa chamada Alto Rio Preto Informatica, que disponibilizou um excelente “case” em seu site. http://www.altoriopreto.com.br/case1_tech2.php

Terceiro Passo: Desenvolvendo o script do firewall e criando o roteamento.

Nosso script do firewall, que é chamado pelo /etc/rc.local, está localizado no nosso exemplo dentro do diretório /etc/firewall, o arquivo será chamado de firewall.sh, onde estão inseridas todas as regras para funcionamento do próprio firewall, quando as regras para balanceamento, utilizando-se a ferramenta (comando) "ip".

O firewall está baseado no iptables, e foi escrito com base na documentação do firewall de exemplo do Márcio Oliveira (marcio@netkraft.com.br), que prove acesso a internet, para uma rede interna, através do squid, mas neste exemplo, não será usado o squid, nem mesmo para a utilização de proxy transparente, por bom senso deixamos apenas o NAT puro.

Ao longo do script existirão comentários sobre as regras, do que o script está executando, mantendo assim uma linha lógica de configuração, simplificando a identificação de comandos.

---------- /etc/firewall/firewall.sh ----------

```
#!/bin/sh
########################################
# Firewall for ADSL Velox Telemar      #
# By Leandro R. - bolinh0@click21.com.br  #
# OBS: Balanceamento de trafego para   #
#      2 links ADSL                    #
########################################

# Legenda
#
# eth0 -> Rede interna (192.168.0.1/24)
# eth1 -> Velox #1     (192.168.1.2/24)
# eth2 -> Velox #2     (192.168.2.2/24)

# Variaveis
DNS1="200.202.93.75"
DNS2="200.202.0.34"

# Carregando os modulos basicos:

echo -n "Carregando os modulos..."
modprobe ip_tables
modprobe iptable_filter
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe iptable_nat
modprobe ip_nat_ftp
```

```
modprobe ipt_LOG
modprobe ipt_state
modprobe ipt_MASQUERADE
echo "                                [OK]"

# Resetando o Firewall:

echo -n "Resetando o firewall..."
iptables -F
iptables -Z
iptables -X
iptables -t nat -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
echo "                                 [OK]"

# Habilitando o roteamento de pacotes:

echo -n "Habilitando o roteamento..."
echo "1" > /proc/sys/net/ipv4/ip_forward
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
echo "                               [OK]"

# Liberando a chain INPUT para o localhost:

echo -n "Liberando acesso do localhost..."
iptables -A INPUT -p ALL -s 127.0.0.1 -i lo -j ACCEPT
iptables -A INPUT -p ALL -s 192.168.0.1 -i lo -j ACCEPT
iptables -A INPUT -p ALL -s 192.168.1.2 -i lo -j ACCEPT
iptables -A INPUT -p ALL -s 192.168.2.2 -i lo -j ACCEPT
echo "                            [OK]"

# Otimizando o firewall:

echo -n "Otimizando o roteamento..."
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
echo "                                [OK]"

# Liberando resposta dos servidores DNS:

echo -n "Liberando servidores DNS..."
iptables -A INPUT -p udp -s 192.168.0.0/24 --sport 53 -d $DNS1 -j ACCEPT
iptables -A INPUT -p udp -s 192.168.0.0/24 --sport 53 -d $DNS2 -j ACCEPT
echo "                               [OK]"

# DHCP Server

echo -n "Liberando servidor DHCP..."
iptables -A INPUT -p udp -s 192.168.0.0/24 --sport 79 -d 192.168.0.1 -j ACCEPT
echo  "                                [OK]"
```

```
# Descartar pacotes fragmentados:

echo -n "Bloqueando pacotes fragmentados..."
iptables -A INPUT -i eth1 -f -j LOG --log-prefix "Pacote fragmentado: "
iptables -A INPUT -i eth1 -f -j DROP
iptables -A INPUT -i eth2 -f -j LOG --log-prefix "Pacote Fragmentado: "
iptables -A INPUT -i eth2 -f -j DROP
echo "                        [OK]"

# Bloqueando ataques do tipo SPOOF de IP:

echo -n "Bloqueando spoofing..."
iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
iptables -A INPUT -i eth1 -s 172.16.0.0/12 -j DROP
iptables -A INPUT -i eth1 -s 192.168.0.0/16 -j DROP
iptables -A INPUT -i eth1 -s 224.0.0.0/4 -j DROP
iptables -A INPUT -i eth1 -s 240.0.0.0/5 -j DROP
echo "                                [OK]"

# Liberando alguns acessos por ping:

echo -n "Liberando acesso por ping..."
iptables -A INPUT -p icmp --icmp-type 8 -i eth0 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
iptables -A INPUT -p icmp -s 192.168.0.0/24 -d 0/0 -j ACCEPT
echo "                             [OK]"

# Regra para SSH: (opcional)

echo -n "Liberando acesso ao SSH..."
iptables -A INPUT -p TCP --dport 22 -j ACCEPT
echo "                               [OK]"

# Regras do FORWARD

# Descarta pacotes invalidos:

echo -n "Descartando pacotes invalidos para reenvio..."
iptables -A FORWARD -m state --state INVALID -j DROP
echo "              [OK]"

# Mantendo conexoes ativas:

echo -n "Manutencao de conexoes ativas..."
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
echo "                         [OK]"
```

```
# Liberando acesso ao DNS para a rede interna (Email):

echo -n "Liberando DNS para rede interna..."

iptables -A FORWARD -s 192.168.0.0/24 -j ACCEPT

iptables -A FORWARD -p udp -s 192.168.0.0/24 -d $DNS1 --dport 53 -j ACCEPT
iptables -A FORWARD -p udp -s 192.168.0.0/24 -d $DNS2 --dport 53 -j ACCEPT
iptables -A FORWARD -p udp -s $DNS1 --sport 53 -d 192.168.0.0/24 -j ACCEPT
iptables -A FORWARD -p udp -s $DNS2 --sport 53 -d 192.168.0.0/24 -j ACCEPT
echo "                        [OK]"

# Fazendo mascaramento de enderecos IP (NAO NAT):
# OBS: essa regra eh mutuamente excludente com a proxima, a do NAT
#      ou seja, escolha uma das duas

echo -n "Habilitando o mascaramento..."
#iptables -t nat -A POSTROUTING -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
echo "                            [OK]"

# Marcando pacotes

echo -n "Marcando pacotes..."
iptables -A PREROUTING -t mangle -s 192.168.0.0/24 -d 0/0 -j MARK --set-mark 3
echo "                            [OK]"

# Desabilitando o filtro de pacotes do martian source
echo -n "Desligando rp_filter..."
for eee in /proc/sys/net/ipv4/conf/*/rp_filter; do
  echo 0 > $eee
done
echo "                            [OK]"

# Definindo regras de balanceamento de Link:
echo -n "Balanceando links velox..."

# velox #1
ip route add 192.168.1.0/24 dev eth1 src 192.168.1.2 table velox1
#ip route add 192.168.0.0/24 via 192.168.0.1 table velox1
ip route add default via 192.168.1.1 table velox1

# velox #2
ip route add 192.168.2.0/24 dev eth2 src 192.168.2.2 table velox2
#ip route add 192.168.0.0/24 via 192.168.0.1 table velox2
ip route add default via 192.168.2.1 table velox2

# setando velox na tabela principal de roteamento
ip route add 192.168.1.0/24 dev eth1 src 192.168.1.2
ip route add 192.168.2.0/24 dev eth2 src 192.168.2.2
```

```
# setando a rota preferencial
ip route add default via 192.168.1.1

# regras das tabelas
ip rule add from 192.168.1.2 table velox1
ip rule add from 192.168.2.2 table velox2

# balanceamento de link
ip rule add fwmark 3 lookup velox prio 3
ip route add default table velox nexthop via 192.168.1.1 dev eth1 weight 1 nexthop via 192.168.2.1
dev eth2 weight 1
#OBS: o comando assima deve ser digitado em uma só linha, não deve ser dado [enter].

# flush no roteamento
ip route flush cache
echo "                                    [OK]"
sleep 3
```

---------- /etc/firewall/firewall.sh ----------

O script ainda libera o acesso para que o nosso serviço de DHCP rodando neste mesmo servidor, possa fornecer o pool de enderços IP para os apartamentos do edifício exemplo do nosso cenário.

No final do script, ou seja, na parte onde é feita as configurações efetivas para o funcionamento do balanceamento dos links, é importante a observação que faz referencia ao comando de balanceamento, que deve ser digitado em somente uma linha, não apertando a tecla [Enter]:

Regra: ip route add default table velox nexthop via 192.168.1.1 dev eth1 weight 1 nexthop via 192.168.2.1 dev eth2 weight 1

Outra importante consideração sobre o script, e a utilização da marcação de pacotes, para que os mesmos fossem roteados pelas tabelas de roteamento das duas conexões ADSL. Primeiramente foi criada uma regra para o firewall, através do iptables, dizendo para marcar todos os pacotes com destino de saída da rede.

Regra: iptables -A PREROUTING -t mangle -s 192.168.0.0/24 -d 0/0 -j MARK --set-mark 3

Em segundo lugar foi utilizada essa marcação nos pacotes, para servir de entrada como uma “Classificação” para a tabela de roteamento (30 Velox).

Regra: ip rule add fwmark 3 lookup velox prio 3

Outro detalhe importante é a configuração da primeira rota padrão, as conexões, mesmo estando balanceadas, neste caso de 1:1 (soma dos links)

precisam de ter uma rota padrão, pois está será por onde o balanceamento deverá ser iniciado, ou seja, a primeira conexão que chegar ao servidor, será transmitida para a internet pela rota padrão, a próxima conexão que for solicitada, será transmitida pelo outro link (balanceamento de 1:1, uma conexão para cada link).

Assim um exemplo seria, o download de um arquivo de 200MBs, se fosse iniciado o download deste arquivo a partir de um gerenciador de downloads, e utilizando-se opções que permitam a divisão do download em outras conexões, seria possível por exemplo "baixar" 100mbs a partir de cada conexão, levando-se em consideração que foi escolhida a opção de dividir o download em duas partes.

Em números: Se o download através de uma conexão está sendo feito a 20kbps, dividindo este download, e utilizando o balanceamento para usar a conexão do outro modem ADSL, é possível que o download possa ser feito ao dobro da velocidade, como no exemplo, 40kbps.

Além do balanceamento este script já prove redundancia de conexões, ou seja se uma conexão cair, ele automaticamente redireciona todo o tráfego apenas para a conexão que continuou ativa.

Recomenda-se o uso de alguma ferramenta que possa monitorar as conexões, uma excelente alternativa é o IpTraf.
http://cebu.mozcom.com/riker/iptraf/download.html

Nele será possível ver as conexões ativas dos usuários, a velocidade real das conexões ADSL, e ver o balanceamento em funcionamento.

Quarto Passo: Simples configuração de um servidor DHCP.

Este é um exemplo de configuração do servidor DHCP no Linux, utilizando o serviço DHCPD, através do arquivo de configuração dhcpd.conf, localizado dentro do diretório /etc.

Não entrarei em detalhes sobre a sua configuração, as informações nele configuradas, mostram, a ativação deste serviço para a rede 192.168.0.X, os endereços por ele distribuidos estão na faixa de .10 a .250, e algumas outras opções muito importantes tambem não poderiam deixar de ser setadas, como a rota padrão e os servidores de DNS.

```
------------------------------------ /etc/dhcpd.conf ------------------------------
#---------------------------------------------
default-lease-time 43200;
max-lease-time 43200;
option domain-name "minharede.intranet";
option domain-name-servers 200.X.X.X, 200.X.X.X;
ddns-update-style ad-hoc;
subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.10      192.168.0.250;
        option routers          192.168.0.1;
        option subnet-mask      255.255.255.0;
        allow unknown-clients;
}
#---------------------------------------------
------------------------------------ /etc/dhcpd.conf ------------------------------
```

Para maiores informações sobre o serviço DHCP, é aconselhavel o uso do "man", pois existem muitas informações e parametros de configurações contidas nele.

Considerações Finais:

Este script está em funcionamento extamente do jeito que foi posto aqui. É claro que milhares de outras configurações podem ser feitas e utilizadas com o balanceamento de links, como por exemplo a divisão de utilização de links por portas, ou aplicações, mas este NÃO é o objetivo deste pequeno artigo, que se resume a demonstrar de forma simplificada o balanceamento de conexões.

Dúvidas ou sugestões – email: bolinh0@click21.com.br

Agradecimento: Douglas Coelho S.

PS: Antes de criticar o trabalho dos outros, pense em fazer melhor.

Leandro R. 03/06/2004