- NAT de entrada:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -d
  192.168.1.100 -j DNAT --to 10.0.0.1
```

A regra anterior diz que tudo que chegar para a porta 80 TCP da máquina 192.168.1.100 será redirecionado para a mesma porta da máquina 10.0.0.1.

- Remover todas as regras de NAT: # *iptables -t nat -F*

- Mais detalhes: http://eriberto.pro.br/iptables.

6. tcpdump

- Mostrar todo o tráfego, sem resolver nomes, na primeira interface listada com # *ifconfig* ou com # *tcpdump -D*:

```
# tcpdump -n
```

- Mostrar todo o tráfego, sem resolver nomes, na *eth1*:

```
# tcpdump -ni eth1
```

- Mostrar todo o tráfego, sem resolver nomes, em qualquer interface:

```
# tcpdump -ni any
```

- Mostrar todo o tráfego que envolva 10.0.0.1:

```
# tcpdump -n host 10.0.0.1
```

- Mostrar todo o tráfego que envolva um MAC:

```
# tcpdump -n ether host 00:00:00:00:10:01
```

- Mostrar todo o tráfego que envolva a porta 80:

```
# tcpdump -n port 80
```

- Mostrar todo o tráfego, incluindo o payload e camada 2, que envolva um IP e uma porta:

```
# tcpdump -nAe host 10.0.0.1 and port 80
```

- Gravar um tráfego: use *-w arquivo.dump*

- Ler um tráfego: use *-r arquivo.dump*

7. Controle de tráfgo (delay pool e HTB)

- O controle de tráfego determina como os pacotes devem circular em uma rede, de forma justa para todos.

- No controle de tráfego a base é decimal. Assim, 1 MB/s (megabyte/segundo) = 1.000 KB/s. Para converter para bits, multiplique por 8. Exemplo: 1 MB/s = 8 Mb/s.

- Delay pool é um método do Squid utilizado para controlar download por rede, subrede ou IP. Mais referências poderão ser vistas em http://bit.ly/delay_pool.

- O delay pool utiliza o conceito de baldes e possui as seguintes classes:

```
1: Apenas um balde limita a rede como um todo.
2: Um balde atua sobre toda a rede e outro considera
   apenas o último octeto do IP.
3: Um balde atua sobre toda a rede. O segundo sobre o
   terceiro octeto. O terceiro sobre o terceiro
   (novamente) e o quarto octeto.
```

- HTB é um disciplina de controle, que pode ser combinada comPRIO, possibilitanto o controle de qualquer tráfego. Isto não dispensa o uso de delay pool. Mais detalhes sobre HTB em http://bit.ly/htb_iptables.

8. Alguns comandos úteis

- Verificar todas as portas servidoras: # *neststat -tunlp*

- Acompanhar todo o tráfego e uso do link: # *iptraf*

- Consultar IP ou nome em DNS:

```
# apt-get install dnsutils
# nslookup www.terra.com.br
# nslookup 200.154.56.80
```

- Especificar o 8.8.8.8 como DNS de consulta:

```
# nslookup www.terra.com.br 8.8.8.8
```

- Verificação da situação da placa de rede e link:

```
# apt-get install ethtool net-tools
# mii-tool
# ethtool eth0
```

9. Referências interessantes

- ***Eriberto (Site):*** http://eriberto.pro.br (especial atenção para os subdiretórios /blog e /wiki)
- ***Linux Advanced Routing & Traffic Control:*** http://lartc.org
- ***NetworkConfiguration:*** http://wiki.debian.org/\NetworkConfiguration
- ***Tcpdump:*** http://eriberto.pro.br/files/guia_tcpdump.pdf

GUIA BÁSICO DE REDES TCP/IP NO DEBIAN GNU/LINUX

Versão 1.2 - 31 de outubro de 2014

http://eriberto.pro.br / eriberto@eriberto.pro.br

Twitter: @eribertomota

4096R/04EBE9EF: 357D CB0E EC95 A01A EBA1 F0D2 DE63 B9C7 04EB E9EF

1. Preâmbulo

Este guia de referência foi criado com o intuito de servir como orientação e checklist básico para a montagem de redes de computadores com **Debian GNU/Linux**. Não se trata de algo avançado. É apenas uma orientação genérica.

2. Configuração de rede IPv4

2.1 Endereço IP, máscara e default gateway

- Arquivo */etc/network/interfaces*. Exemplo:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
      address 10.0.0.1
      netmask 255.0.0.0
      gateway 10.0.0.100
      up echo Aguarde...; /usr/local/bin/rotas.sh
```

A linha auto, caso exista, provocará a ativação da interface juntamente com o boot do sistema operacional. A linha gateway é opcional e refere-se ao default gateway. Deverá ser colocada no bloco referente à mesma rede (ex: 10.0.0.100 pertence à rede 10.0.0.0/8). O parâmetro up é opcional. Tudo que for colocado depois dele será executado após a inicialização da placa de rede. Excelente opção para ativar rotas e regras de filtros de pacotes. Também há o parâmetro down. Poderá haver várias linhas com up ou down.

A configuração de loopback é obrigatória.

- Poderão ser criadas aliases de IP (interfaces virtuais). Com isso, a mesma interface real poderá receber dois ou mais IPs diferentes (virtuais). Bastará adicionar um novo bloco de configuração com *interface:apelido*. Ex:

```
auto eth1:teste
iface eth1:teste inet static
      address 192.168.1.1
      netmask 255.255.255.0
```

- Depois de editado o arquivo de configuração, utilize o seguinte comando para ativar uma determinada interface:

```
# ifup eth1
```


- Caso deseje reiniciar totalmente uma interface, faça o seguinte:

```
# ifconfig eth0 0.0.0.0
# ifdown eth0
# ifup eth0
```

As interfaces virtuais (*eth1:teste*, por exemplo) deixarão de existir caso as reais saiam do ar. Ainda, o antigo comando */etc/init.d/networking (start|stop|restart)* está em desuso e não tem mais o pleno controle sobre as interfaces de rede. Utilize sempre *ifup* e *ifdown*.

- Obs: DESABILITE o network-manager nas máquinas.

2.2 Servidores DNS

- Os servidores DNS deverão ser citados dentro de */etc/resolv.conf*. Exemplo:

```
nameserver 8.8.8.8
nameserver 208.67.222.222
```

Os endereços IP mostrados pertencem, respectivamente, aos DNS públicos do Google e do projeto OpenDNS. Você poderá utilizá-los normalmente para resolver algo na Internet. Na verdade, as duplas de endereços são: 8.8.8.8 e 8.8.4.4 (Google), além de 208.67.220.220 e 208.67.222.222 (OpenDNS). Mais detalhes em http://eriberto.pro.br/blog/?p=849.

2.3 Rotas estáticas adicionais

- As rotas estáticas adicionais (as que não são default gateway) poderão ser configuradas com o comando *route*. Exemplos:

```
# route add -net 172.16.0.0/16 gw 10.0.0.8
```

No comando anterior foi dito que será utilizado o gateway 10.0.0.8 para que cheguemos à rede 172.16.0.0/16.

- Se for o caso, as rotas criadas poderão ser inseridas em um *up* dentro do arquivo */etc/network/interfaces, como mostrado no item 2. deste guia.*

3. Configuração de bridge

- Pacote necessário:

```
# apt-get install bridge-utils
```

- Arquivo */etc/network/interfaces*. Exemplo:

```
auto lo
iface lo inet loopback
```



```
auto eth0
iface eth0 inet static
      address 127.0.0.2
      netmask 255.0.0.0

auto eth1
iface eth1 inet static
      address 127.0.0.3
      netmask 255.0.0.0

auto br0
iface br0 inet static
      address 10.0.0.1
      netmask 255.0.0.0
      gateway 10.0.0.100
      bridge_ports eth0 eth1
```

Foram atribuídos endereços da rede loopback às interfaces eth0 e eth1. A interface br0 (bridge), foi criada com endereço IP e default gateway. Depois, as interfaces eth0 e eth1 foram associadas. Assim, será possível acessar a bridge por SSH ou outro método. No entanto, a bridge não precisa de IP acessível:

```
auto br0
iface br0 inet static
      address 127.0.0.4
      netmask 255.0.0.0
      bridge_ports eth0 eth1
```

- O comando # *brctl show* mostra a situação da bridge.
- Mais detalhes: http://bit.ly/bridge_debian.

4. IP forward

O IP forward é um procedimento que permite a passagem de dados entre placas de rede em uma mesma máquina. Ele é obrigatório em roteadores e elementos de firewall que atuam na camada 3 (OSI). Para ativar, edite o arquivo */etc/sysctl.conf* e descomente a linha:

```
net.ipv4.ip_forward=1
```

A seguir, aplique o comando: # */etc/init.d/procps start*

5. NAT com Iptables

- NAT de saída:

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

A regra anterior diz que tudo o que for sair pela interface eth1 sofrerá NAT, recebendo o IP de tal interface.