

Roteamento avançado e controle de banda em Linux

Hélio Loureiro

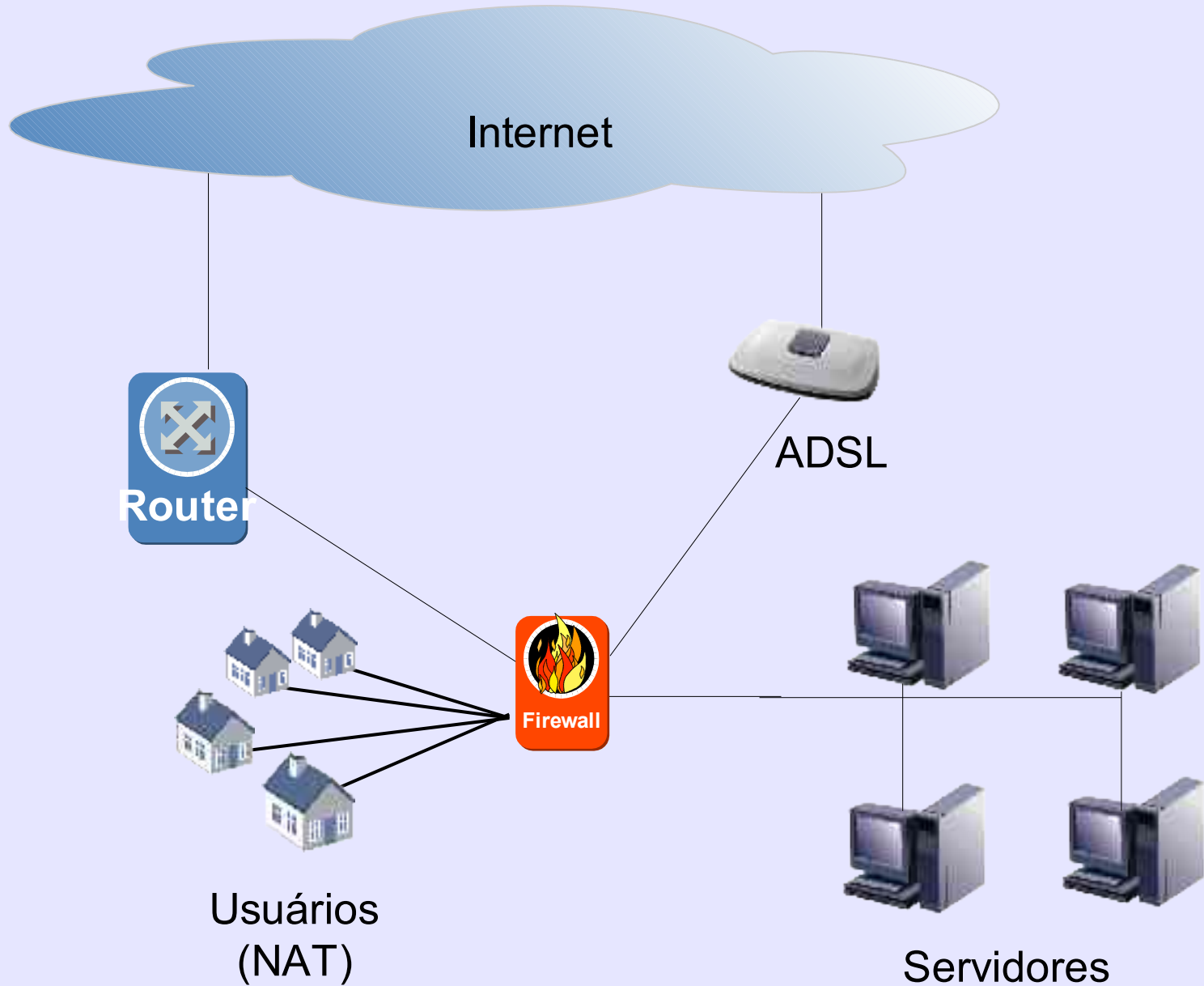
<helio@loureiro.eng.br>

- Roteamento avançado
 - Sintaxe
 - Exemplo
- Controle de banda
 - Sintaxe
 - Exemplo

NOTA: os exemplos são baseados na distribuição Debian mas funcionam similarmente em todas as demais.

Cenário típico

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda








Busca de informações

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda



"policy routing" <O.S.>

- Linux - iptables/ipchains/iproute2 - 8.030 links
- FreeBSD - ipfw/ipfw2/ipfilter - 2.290 links 
- OpenBSD - ipfilter/pf - 1.430 links
- Solaris - ipfilter - 2.048 links 
- NetBSD - ipfilter - 1.320 links 
- SCO (Unixware) - ? - 1.440 links 
- Windows - ? - 5.200 links 

O termo “policy routing” é utilizado em roteadores, enquanto que “source routing” em Linux/BSD”.

Linux

- › iptables - sintaxe muito flexível (complexa)
- › iptables - difícil padronização para criação script
- › iproute2 - fácil configuração para roteamento
- › iproute2 - estável!

OpenBSD

- › pf - sintaxe simples (BSD)
- › pf - roteamento através dele
- › pf - recém lançado na versão 3.0



17º REUNIÃO – GTER

Roteamento avançado

Controle de banda

```
block in log all
scrub in all
pass out all keep state
pass out inet proto tcp from any to any keep state
pass in quick on $EXT proto tcp from $HELIO to $FW \
    port 22 flags S/SA keep state
pass in quick proto udp from any to any port 53 keep state
pass in on $EXT inet proto tcp from any to $SERVER \
    port {25,80,110,143,443} flags S/SA modulate state
pass in on $EXT proto udp from any port 53 to $FW keep state
pass in on xl0 fastroute from 192.168.0.0/24 to $DMZ keep
state
pass in on xl0 fastroute from 192.168.0.0/24 to $EXTERNAL keep
state
pass in on xl0 route-to xl3:200.100.10.1 from \
    192.168.0.0/24 to any keep state
```

Sistema travava...

Preparando o kernel

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda

```
[*] Prompt for development and/or incomplete  
code/drivers
```

```
---
```

```
[*] TCP/IP Networking
```

```
[*] Networking packet filtering (replaces ipchains)
```

```
[*] Networking packet filtering debugging
```

```
[*] Socket Filtering
```

```
---
```

```
[*] IP: advanced router
```

```
[*] IP: policy routing
```

```
[*] IP: use netfilter MARK value as routing key
```

```
[*] IP: fast network address translation
```

```
[*] IP: equal cost multipath
```

```
[*] IP: use TOS value as routing key
```

```
[*] IP: verbose route monitoring
```

```
[*] IP: large routing tables
```

```
[*] IP: GRE tunnels over IP
```

Já disponível no kernel
2.4.20 ou superior

- Solução desenvolvida para o kernel 2.4.
- Substitui os comandos `arp`, `ifconfig` e `route`.
- Sintaxe semelhante à cli de roteadores.
- Permite criar regras de roteamento.
- Não interage com os comandos legados.
- Inclui o programa de controle de banda (***tc***).
- Usa o sistema de filtros e filas.

apt-get install iproute

Usado para verificar e/ou configurar o endereço físico (MAC) das interfaces de rede. Aceita as opções `show` e `set`.

```
router:~# ip link show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:04:75:7a:73:63 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:04:75:7a:73:8e brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:04:75:7a:73:31 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
```

Usado para interagir com as interfaces de rede. Aceita as opções `list`, `add` e `del` entre outras.

```
router~# ip addr list
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:10:5a:9b:1e:fd brd ff:ff:ff:ff:ff:ff
   inet 192.168.254.200/24 brd 192.168.254.255 scope global eth0
router~# ip addr add 192.168.253.200/24 dev eth0
router~# ip addr list dev eth0
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 00:10:5a:9b:1e:fd brd ff:ff:ff:ff:ff:ff
   inet 192.168.254.200/24 brd 192.168.254.255 scope global eth0
   inet 192.168.253.200/24 scope global eth0
router~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:10:5A:9B:1E:FD
          inet addr:192.168.254.200  Bcast:192.168.254.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

ip route

17º REUNIÃO – GTER Roteamento avançado Controle de banda

Usado para verificar e/ou configurar as rotas estáticas da rede.
Aceita as opções `list`, `flush`, `add`, e `del` entre outros.

```
router:~# ip route list
200.1.2.0/26 dev eth2 proto kernel scope link src 200.1.2.20
200.100.10.0/26 dev eth3 proto kernel scope link src 200.100.10.56
10.0.0.0/24 dev eth1 proto kernel scope link src 10.0.0.254
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.254
default via 200.1.2.1 dev eth2
router:~# ip route add nat 192.168.0.100 via 200.1.2.20
```

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). Uma entrada no arquivo mas sem regra definida não é apresentada no comando "ip rule list". Para forçar o kernel a ler a nova entrada, o comando "ip route flush cache" é necessário.

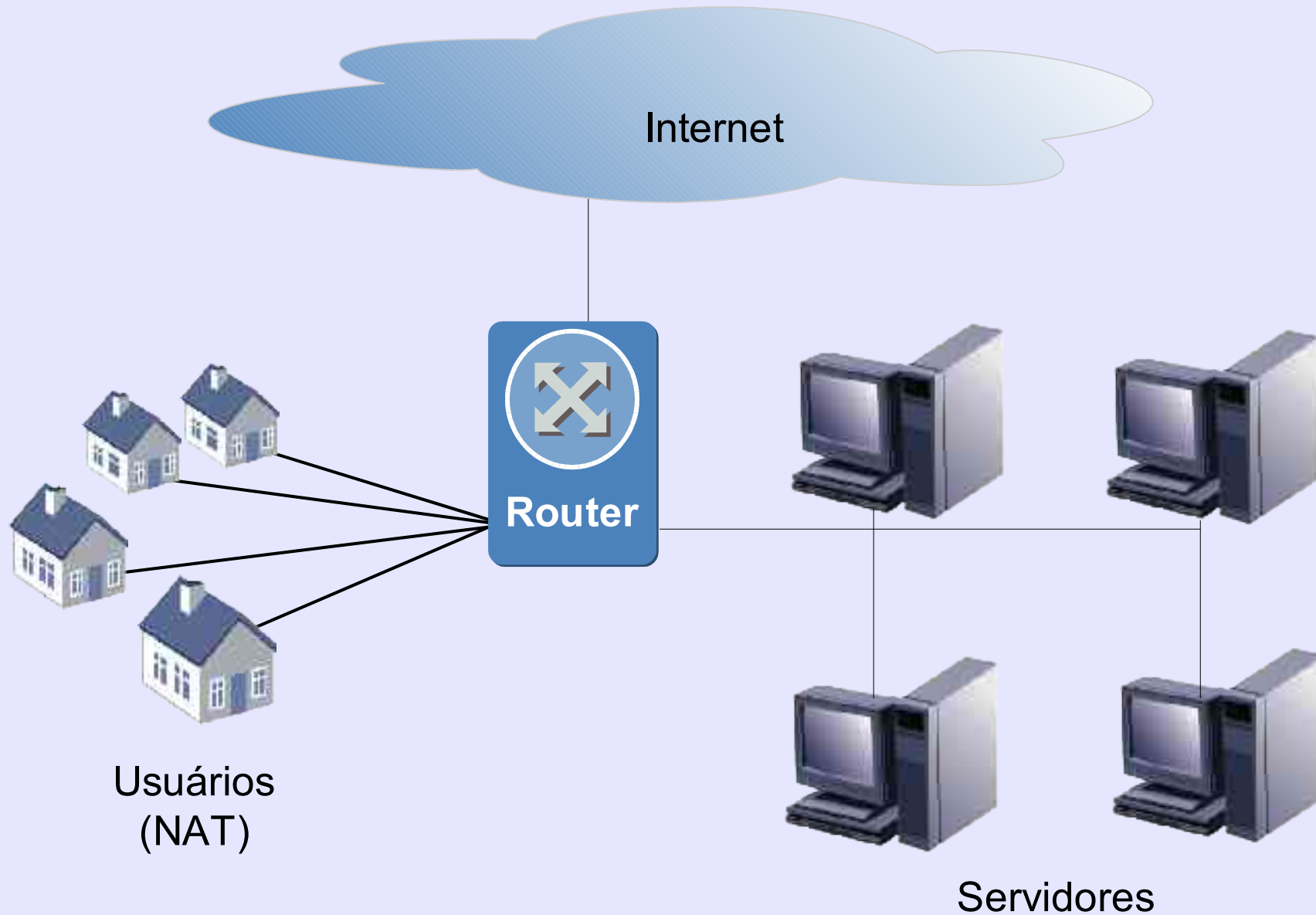
```
router ~# echo "200 dmznet" >> /etc/iproute2/rt_tables
router ~# cat /etc/iproute2/rt_tables
# reserved
255    local
254    main
253    default
0     unspec
# local
1     inr.ruhep
200    dmznet
```

Usado para criar regras específicas de roteamento. Existem algumas tabelas iniciais que não podem ser removidas: **local** (loopback), **main** e **default**. Cria-se uma ou mais tabelas para o roteamento desejado. No caso apresentado, somente a tabela **dmznet** foi adicionada, deixando todos os demais dentro da tabela **main**. Aceita as opções `list`, `add` e `del`.

```
router:~# ip rule list
0:      from all lookup local
32762:  from 10.0.0.11 lookup dmznet
32763:  from 192.168.0.0/24 to 200.1.2.0/26 lookup dmznet
32764:  from 200.100.10.0/26 lookup dmznet
32765:  from 192.168.0.0/24 lookup dmznet
32766:  from all lookup main
32767:  from all lookup default
```

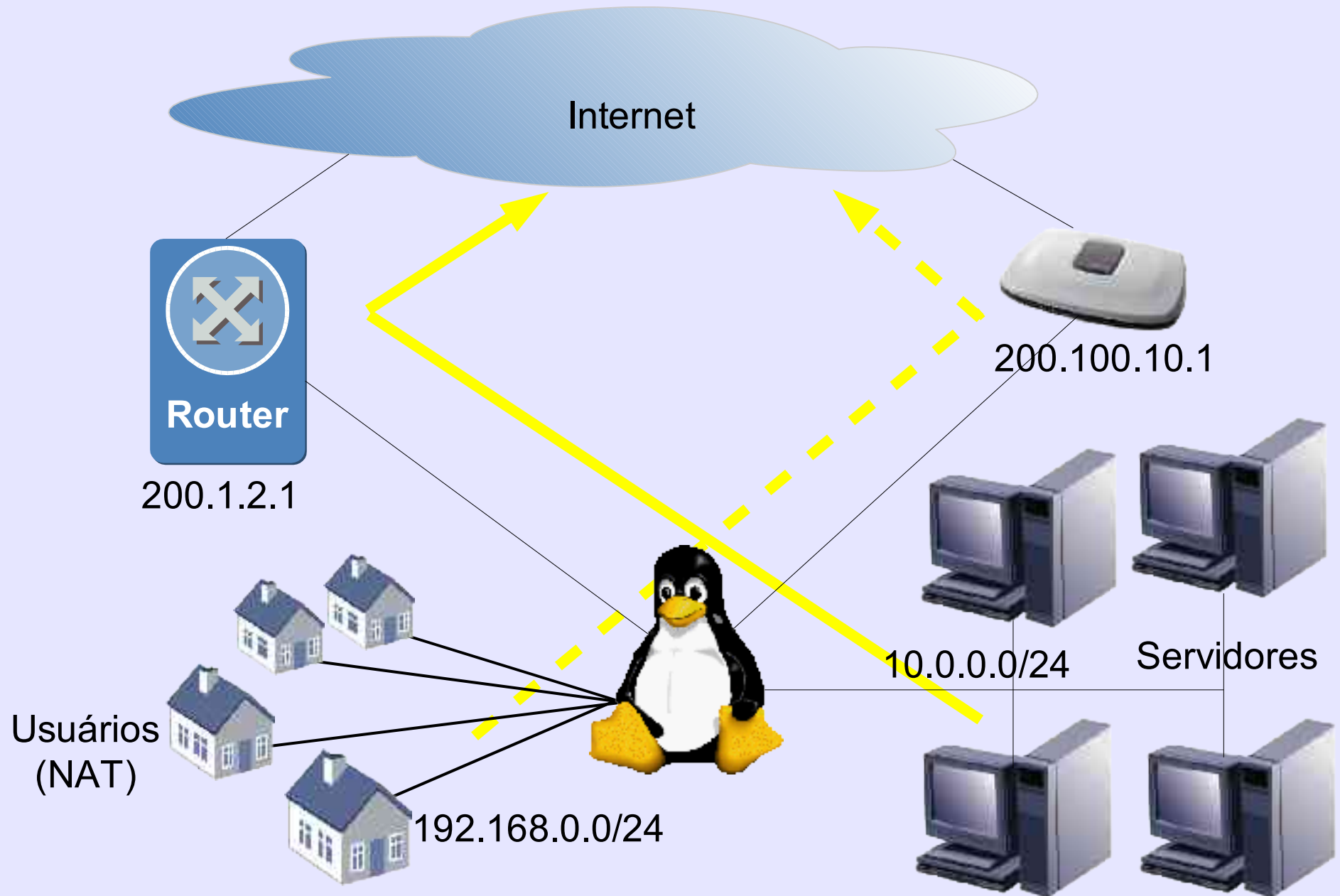
Situação inicial (uma saída)

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda



Situação desejada

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda



/etc/init.d/iproute

17º REUNIÃO – GTER Roteamento avançado Controle de banda

```
#!/bin/sh
ip route add default via 200.100.10.1          table dmznet
ip rule add from 192.168.0.0/24                table dmznet
ip route add 192.168.0.0/24 via 192.168.0.254 table dmznet
ip rule add from 200.100.10.0/26               table dmznet
ip addr add 200.1.2.3/26 dev eth2
ip addr add 200.1.2.4/26 dev eth2
ip addr add 200.1.2.5/26 dev eth2
ip addr add 200.1.2.7/26 dev eth2
ip addr add 200.1.2.14/26 dev eth2
ip addr add 200.1.2.15/26 dev eth2
ip addr add 200.1.2.27/26 dev eth2
ip addr add 200.1.2.62/26 dev eth2
ip rule add from 192.168.0.0/24 to 200.1.2.0/26 table dmznet
ip route add 200.1.2.0/26 via 200.1.2.20      table dmznet
ip route add 10.0.0.0/24 via 10.0.0.254       table dmznet
ip rule add from 10.0.0.11/32                  table dmznet
```

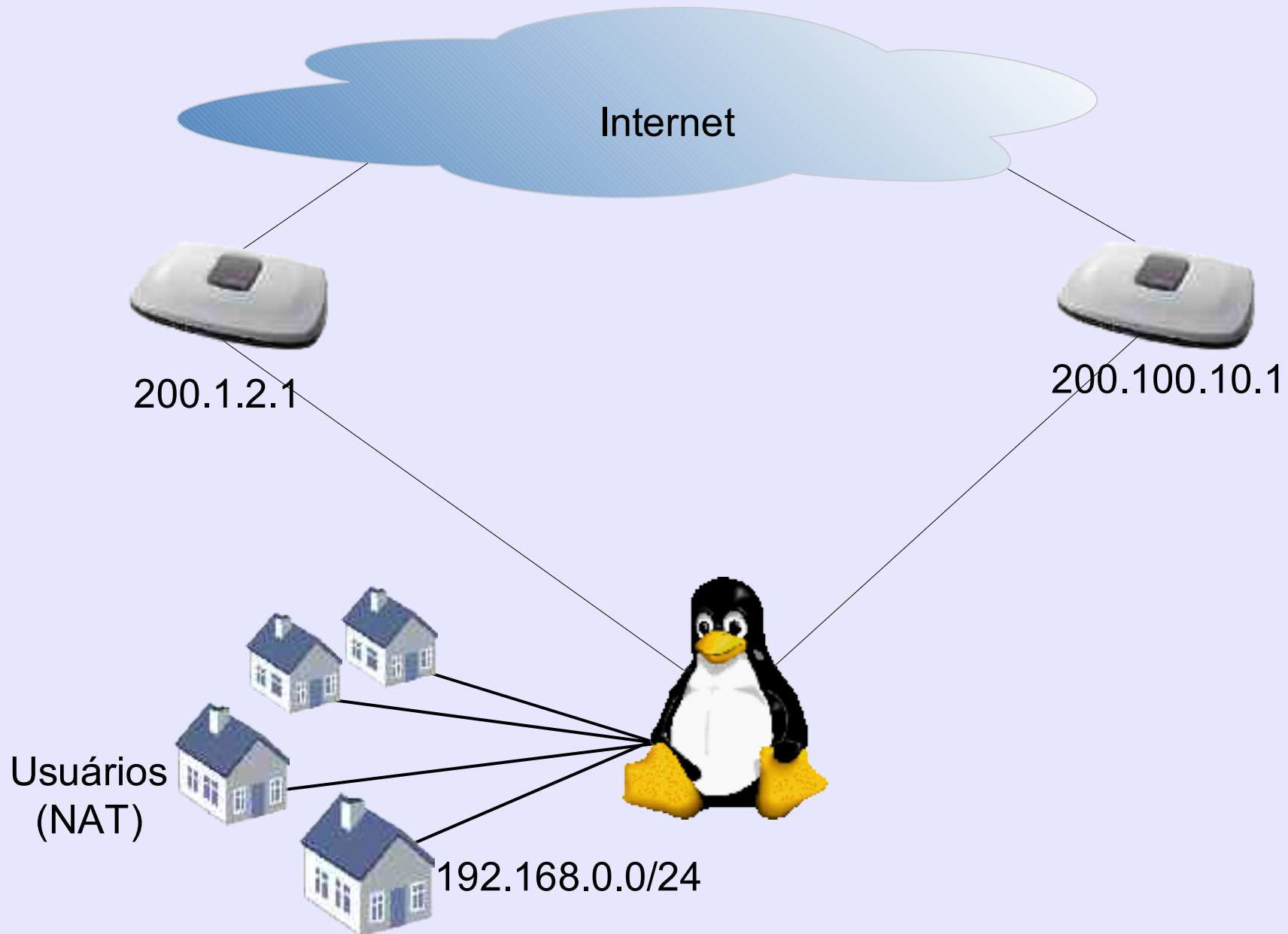

Alterando rotas

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda

```
#!/bin/sh
case $1 in
  start|adsl)
    echo "Roteando Intranet pelo link ADSL"
    ip route del default via 200.1.2.1 table dmznet
    ip route add default via 200.100.10.1 table dmznet
    echo "Iniciando regras de firewall"
    /etc/init.d/firewall start
    ;;
  stop|frame-relay)
    echo "Roteando Intranet pelo link FR"
    ip route del default via 200.100.10.1 table dmznet
    ip route add default via 200.1.2.1 table dmznet
    echo "Desligando regras de firewall"
    /etc/init.d/firewall stop
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  *)
    echo "Use: $0 {start|adsl|stop|frame-relay|restart}"
```

Balanceamento de carga

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda



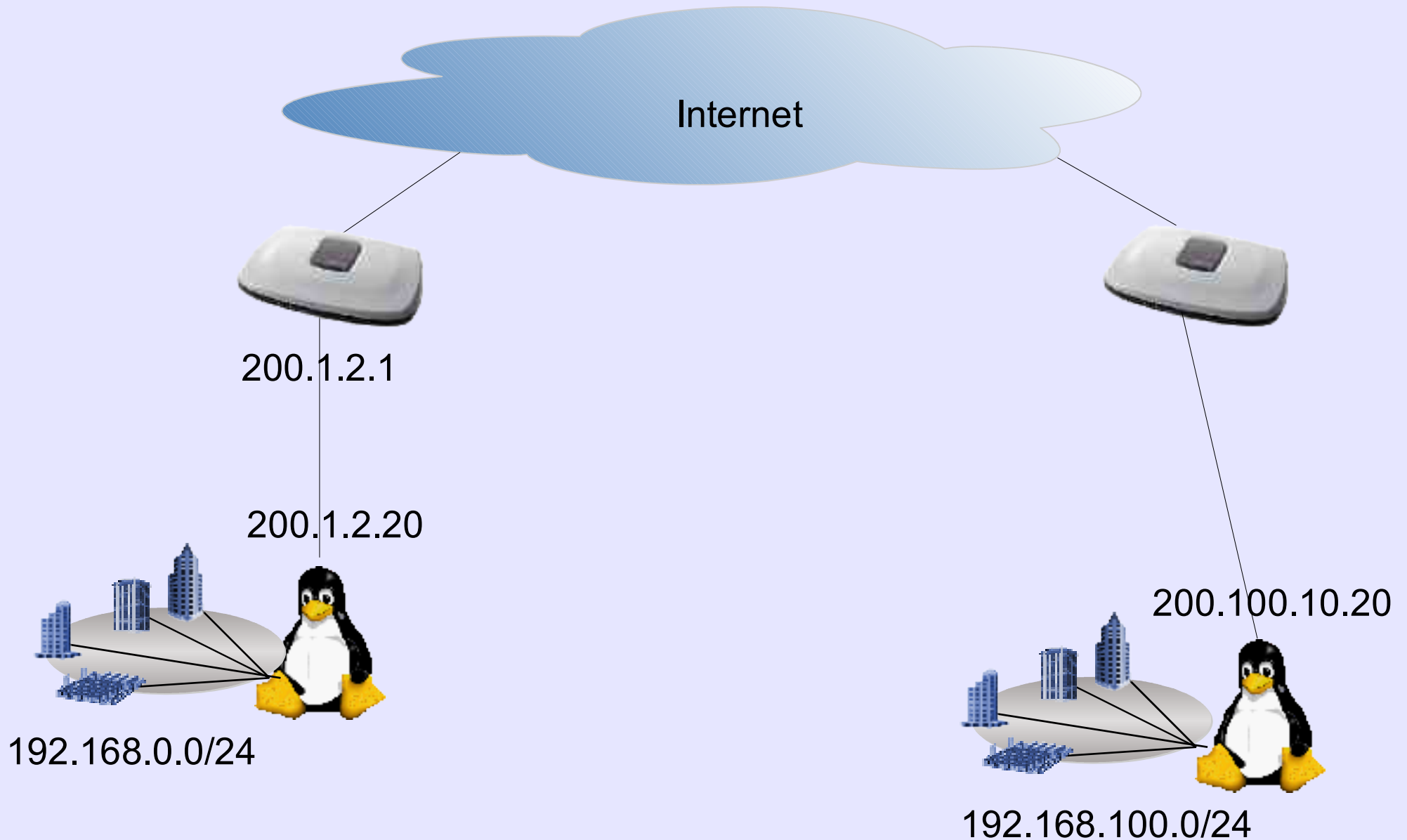
Atribuindo peso às rotas

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda

```
ip route add default scope global nexthop \  
  via 200.100.10.1 dev eth0 weight 1 \  
  nexthop via 200.1.2.1 dev eth1 weight 1
```

Túnel GRE

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda



Túnel GRE

17º REUNIÃO – GTER
Roteamento avançado
Controle de banda

```
ip tunnel add netgre mode gre remote 200.100.10.20 \  
local 200.1.2.20 ttl 255  
ip link set netgre up  
ip addr add 10.0.1.1 dev netgre  
ip route add 192.168.100.0/24 dev netgre
```

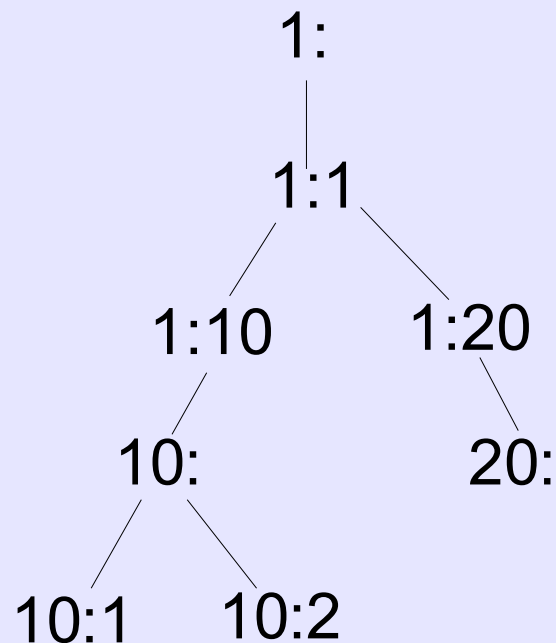
Controle de banda

- Dois tipos básicos: classless e classfull.
- Ambos fazem re-priorização de pacotes.
- Permitem configuração de parâmetros como:
latency, burst, rate, peakrate, etc.
- Em geral funcionam somente na interface *egress* Do sistema (existe uma classe específica para a *Ingress*).
- Pode ser utilizado para dar maior prioridade por host ou por serviço ou ambos.

- Faz controle sobre interface inteira.
- Não permite sub-divisões de classe.
- Configura o comportamento da fila da interface.
- Ex: pfifo_fast, TBF (Token Bucket Filter) e SQF (Stochastic Fairness Queueing).

```
tc qdisc add dev ppp0 root tbf rate 220kbit
```


- Utiliza estrutura de árvore.
- Permite criação de tipos de tráfego com limitação por classe (classless) ou por filho.
- Permite garantia de banda mínima.
- Permite compartilhamento de banda.



Preparando o kernel

17º REUNIÃO – GTER Roteamento avançado Controle de banda

Networking options ---->

QoS and/or fair queueing ---->

[*] QoS and/or fair queueing

<M> HTB packet scheduler

<M> CBQ packet scheduler

<M> CSZ packet scheduler

<M> The simplest PRIQ pseudoscheduler

<M> RED queue

<M> TEQL queue

<M> TBF queue

<M> GRED queue

<M> Diffserv field marker

<M> Ingress Qdisc

[*] QoS support

[*] Rate estimator

[*] Packet classifier API

<M> TC index classifier

<M> Routing table based classifier

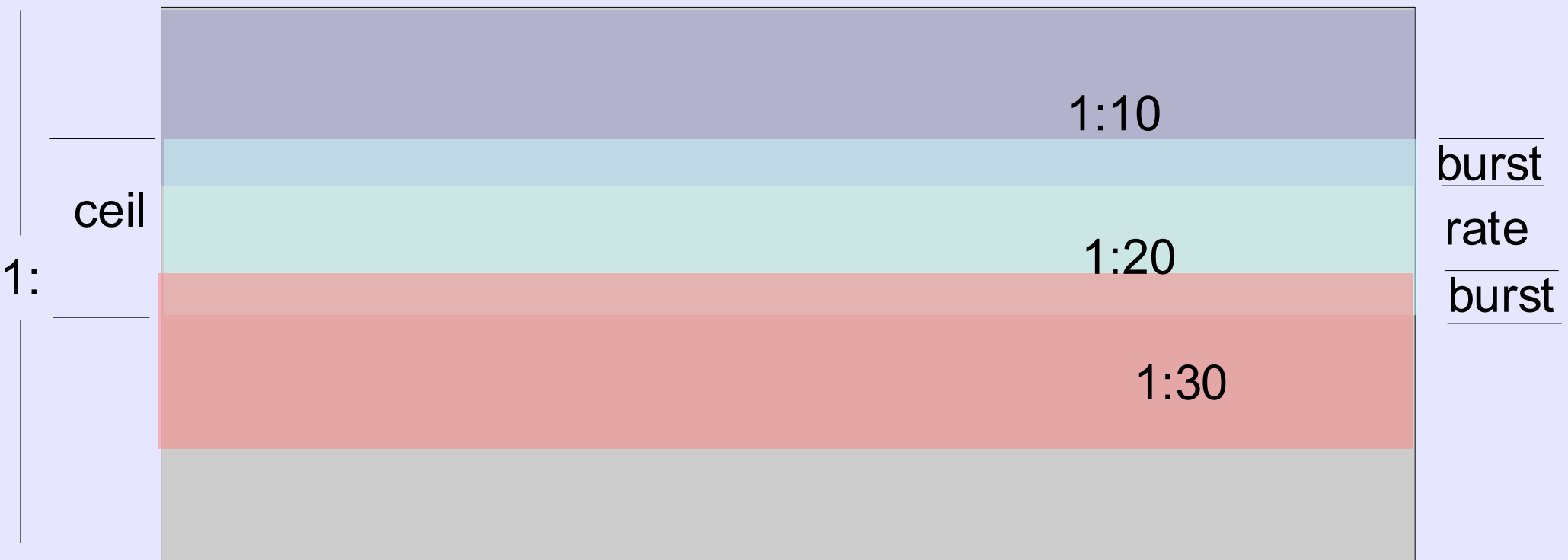
<M> Firewall based classifier

<M> U32 classifier

<M> Special RSVP classifier

[*] Traffic policing (needed for
in/egress)

- Hierarchical Token Bucket.
- Controle preciso (diferente do CBQ).
- Comportamento semelhante ao ALTQ (BSDs).
- Geralmente utiliza outra disciplina de filas dentro de suas classes filhas.



Exemplo

17º REUNIÃO – GTER Roteamento avançado Controle de banda

```
tc qdisc add dev eth0 root handle 1: htb default 1000
tc class add dev eth0 parent 1: classid 1:1 htb \
  rate 256 kbit ceil 256 kbit
tc class add dev eth0 parent 1:1 classid 1:10 htb \
  rate 56 kbit ceil 128 kbit burst 6k
tc class add dev eth0 parent 1:1 classid 1:1000 htb \
  rate 32 kbit ceil 56 kbit burst 6k
tc qdisc add dev eth0 parent 1:10 handle 10: sfq pertub 10
tc qdisc add dev eth0 parent 1:1000 handle 1000: sfq \
  pertub 10
tc filter add dev eth0 parent 1: protocol ip prio 1 \
  u32 match ip src $IP flowid 1:1
tc filter add dev eth0 parent ffff: protocol ip prio 50 \
  u32 match ip src $IP police rate 32 kbit burst 6k \
  drop flowid :1
```

Roteamento avançado com o Linux; Allan Edgard Silva Freitas <allan@cefetba.br>; News Generation; Boletim bimestral sobre tecnologia de redes; 4 de fevereiro de 2002 | volume 6, número 1; http://www.rnp.br/newsgen/0201/roteamento_linux.html

Linux Advanced Routing & Traffic Control HOWTO; Bert Hubert e outros; Netherlabs BV <bert.hubert@netherlabs.nl>; <http://lartc.org/howto/index.html>

IP Command Reference; Alexey N. Kuznetsov;
</usr/share/doc/iproute-2.4.7/ip-cref.ps>

***Agradecimentos e
perguntas????***

**17º REUNIÃO – GTER
Roteamento avançado
Controle de banda**

F I M