

### Atividade 02

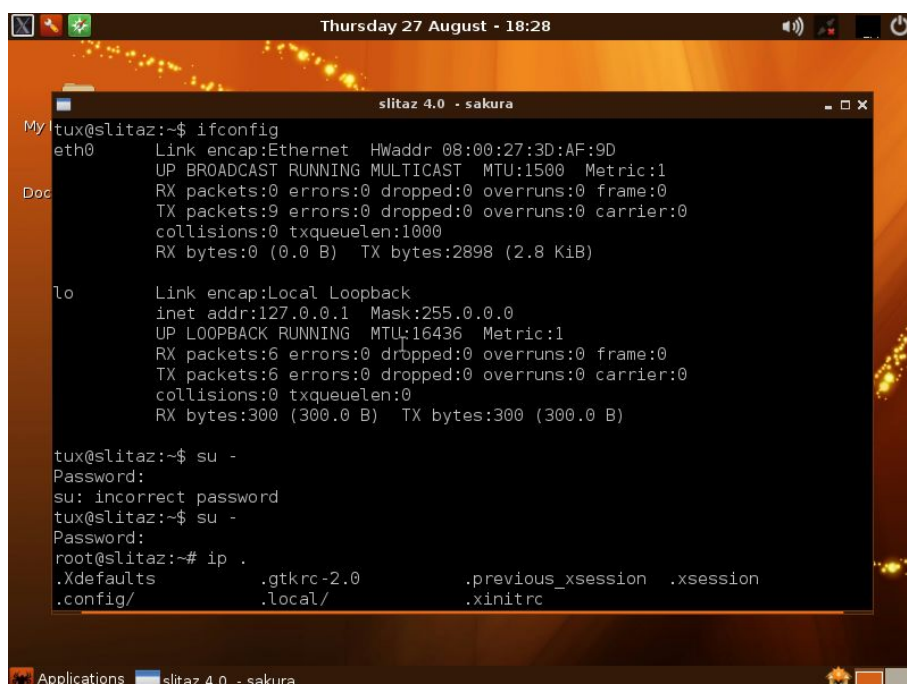
- 2) Após iniciar todos os *hosts* e *routers* virtuais através do *VirtualBox*, foi possível a obtenção das figuras mostradas a seguir, através dos comandos *ping* e dos programas *Wireshark* e *Windbox*. A primeira imagem, logo abaixo, mostra a configuração inicial do roteador 10.0.1.22 (*ip address print*) e, conforme esperado, somente a interface *ether5* está ativada (*interface print*).

```
u
tdown
aug/27/2015 17:37:29 system,error,critical router was rebooted without proper sh
u
tdown
aug/27/2015 17:55:45 system,error,critical login failure for user \1B[admin via
l
local

[admin@MikroTik] > ip addr
[admin@MikroTik] /ip address> print
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 10.0.1.22/23 10.0.0.0 ether5
[admin@MikroTik] /ip address> ..
[admin@MikroTik] /ip> ..
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
# NAME TYPE ACTUAL-MTU L2MTU
0 X ether1 ether 1500
1 X ether2 ether 1500
2 X ether3 ether 1500
3 X ether4 ether 1500
4 R ether5 ether 1500
[admin@MikroTik] > _
```

Figura 1: Configuração do roteador virtual *MikroTik*

A seguir, foi necessária a configuração do sistema operacional do host virtual, cujo endereço IP da interface *ethernet* deve ser alterado para 10.0.0.101. A imagem 2 representa o estado das interfaces desse host logo que ele é iniciado e, como pode ser observado, o endereço da interface *eth0* não está corretamente configurado.



```
Thursday 27 August - 18:28

slitaz 4.0 - sakura

My tux@slitaz:~$ ifconfig
eth0      Link encap:Ethernet  Hwaddr 08:00:27:3D:AF:9D
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:2898 (2.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6 errors:0 drbpped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:300 (300.0 B)  TX bytes:300 (300.0 B)

tux@slitaz:~$ su -
Password:
su: incorrect password
tux@slitaz:~$ su -
Password:
root@slitaz:~# ip .
.Xdefaults      .gtkrc-2.0      .previous_xsession .xsession
.config/        .local/        .xinitrc
```

Figura 2: Configuração inicial do *host* 10.0.0.101

Os comandos `ifconfig eth0 10.0.0.101 netmask 255.255.254.0` e `route add default gw 10.0.1.26` corrigem a configuração inicial, modificando a respectiva interface e adicionando uma nova rota ao roteador conectado à Internet, conforme mostrado na figura 3. Em relação a esta nova rota, destaca-se que a sua máscara de rede vale 0.0.0.0/0, isto é, todos os bits valem 0. Assim, caso o endereço a ser procurado na tabela de roteamento não for encontrado, tal pacote será enviado para 10.0.1.26, uma vez que, após a aplicação da máscara, o endereço é transformado em 0.0.0.0 (operação lógica AND com 0.0.0.0).

```

My root@slitaz:~# ip .
. defaults      . gtkrc-2.0      . previous_xsession . xsession
. config/      . local/
. xinitrc
Doc root@slitaz:~# ifconfig eth0 10.0.0.101 netmask 255.255.254.0
ifconfig: SIOCSIFADDR: No such device
root@slitaz:~# ifconfig eth0 10.0.0.101 netmask 255.255.254.0
root@slitaz:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:3D:AF:9D
          inet addr:10.0.0.101  Bcast:10.0.0.1.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:8372 (8.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:300 (300.0 B)  TX bytes:300 (300.0 B)

root@slitaz:~# route

Kernel IP routing table
Destination: Gateway      Genmask         Flags Metric Ref    Use Iface
127.0.0.1    *                255.255.255.255 UH      0      0      0 lo
10.0.0.0     *                255.255.254.0  U      0      0      0 eth0
root@slitaz:~# route add default gw 10.0.1.26
root@slitaz:~# ping 10.0.1.24
PING 10.0.1.24 (10.0.1.24): 56 data bytes
64 bytes from 10.0.1.24: seq=0 ttl=64 time=0.817 ms
64 bytes from 10.0.1.24: seq=1 ttl=64 time=0.439 ms
64 bytes from 10.0.1.24: seq=2 ttl=64 time=0.469 ms
64 bytes from 10.0.1.24: seq=3 ttl=64 time=0.480 ms
^C
--- 10.0.1.24 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.439/0.551/0.817 ms
root@slitaz:~# route

Kernel IP routing table
Destination: Gateway      Genmask         Flags Metric Ref    Use Iface
127.0.0.1    *                255.255.255.255 UH      0      0      0 lo
10.0.0.0     *                255.255.254.0  U      0      0      0 eth0
default      10.0.1.26        0.0.0.0         IUG      0      0      0 eth0

```

Figura 3: Configuração corrigida do *host* 10.0.0.101

Uma vez todos os equipamentos devidamente configurados, a execução de *ping* nos *hosts* 10.0.0.101 e 10.0.0.200 resulta em

```

My root@slitaz:~# ping 10.0.0.200 -c 10
PING 10.0.0.200 (10.0.0.200): 56 data bytes
64 bytes from 10.0.0.200: seq=0 ttl=128 time=0.568 ms
64 bytes from 10.0.0.200: seq=1 ttl=128 time=0.619 ms
64 bytes from 10.0.0.200: seq=2 ttl=128 time=0.601 ms
64 bytes from 10.0.0.200: seq=3 ttl=128 time=0.614 ms
64 bytes from 10.0.0.200: seq=4 ttl=128 time=0.609 ms
64 bytes from 10.0.0.200: seq=5 ttl=128 time=0.619 ms
64 bytes from 10.0.0.200: seq=6 ttl=128 time=0.625 ms
64 bytes from 10.0.0.200: seq=7 ttl=128 time=0.587 ms
64 bytes from 10.0.0.200: seq=8 ttl=128 time=0.590 ms
64 bytes from 10.0.0.200: seq=9 ttl=128 time=0.585 ms

--- 10.0.0.200 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.568/0.601/0.625 ms
root@slitaz:~#

C:\WINDOWS\system32\cmd.exe
Configuração de IP de Windows

Adaptador Ethernet Conexão Local:
  Sufixo DNS específico de conexão . . : 10.0.0.200
  Endereço IP . . . . . : 10.0.0.101
  Máscara de sub-rede . . . . . : 255.255.254.0
  Gateway padrão . . . . . : 10.0.1.26

C:\Documents and Settings\advin\ping 10.0.0.101
Disparando contra 10.0.0.101 com 32 bytes de dados:
Resposta de 10.0.0.101: bytes=32 tempo=1ms TTL=64
Resposta de 10.0.0.101: bytes=32 tempo=1ms TTL=64
Resposta de 10.0.0.101: bytes=32 tempo=1ms TTL=64
Resposta de 10.0.0.101: bytes=32 tempo=1ms TTL=64
Resposta de 10.0.0.101: bytes=32 tempo=1ms TTL=64

Estatísticas do Ping para 10.0.0.101:
  Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
  Aproximando um número redondo de vezes em milissegundos:
  Mínimo = 0ms, Máximo = 0ms, Média = 0ms

```

Figura 4: Ping nos *hosts* 10.0.0.101 e 10.0.0.200

Um remarque importante a ser destacado é o parâmetro TTL, *Time to Live*, que nos permite detectar se um pacote passou por um roteador ou não, já que toda eventual passagem por um roteador decrementa o valor deste atributo. Dessa forma, se o TTL de um pacote vale 128, então tal pacote não passou por nenhum *router*. Esse atributo é usado assim para evitar o tráfego de pacotes ‘perdidos’ na rede.

Enfim, as figuras a seguir demonstram a utilização dos programas *Wireshark* e *Windbox*. Para o primeiro, foram filtradas os pacotes ICMP, que representam os pacotes enviados através do comando *ping*, e ARP, cuja utilização foi bastante discutida na atividade anterior (atualização da tabela ARP de um determinado *host*). O *Windbox* foi conectado ao roteador 10.0.1.25 e as informações referentes à licença, interfaces e endereço IP foram mostradas. Conforme esperado, o estado das interfaces é o mesmo daquele mostrado na figura 1.

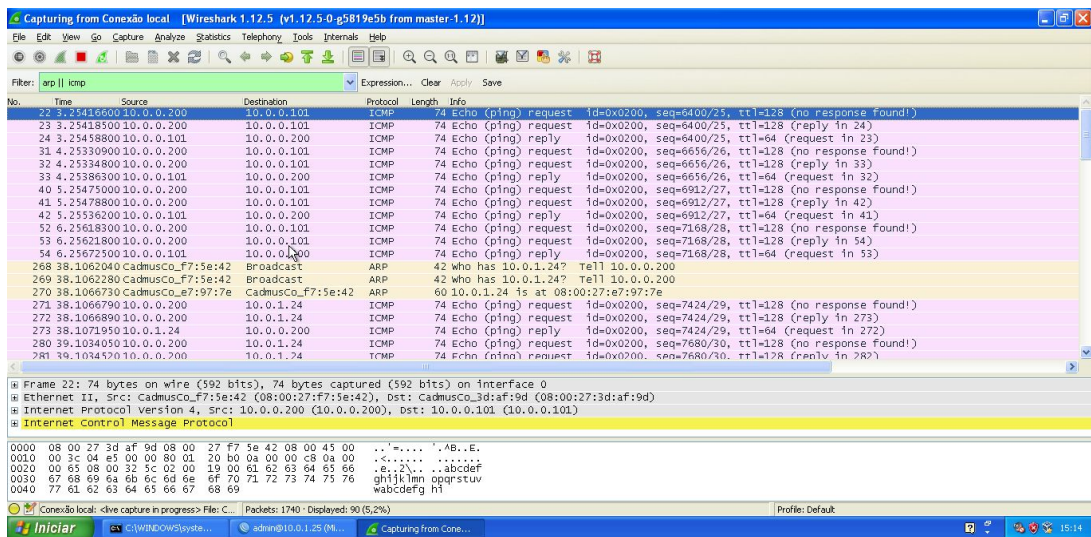


Figura 5: Captura do tráfego do host 10.0.0.200

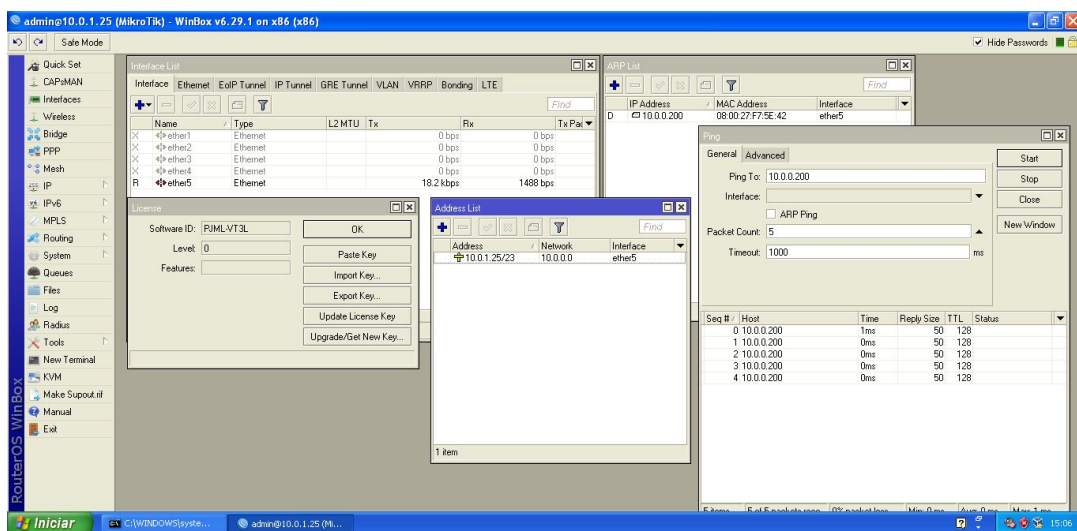
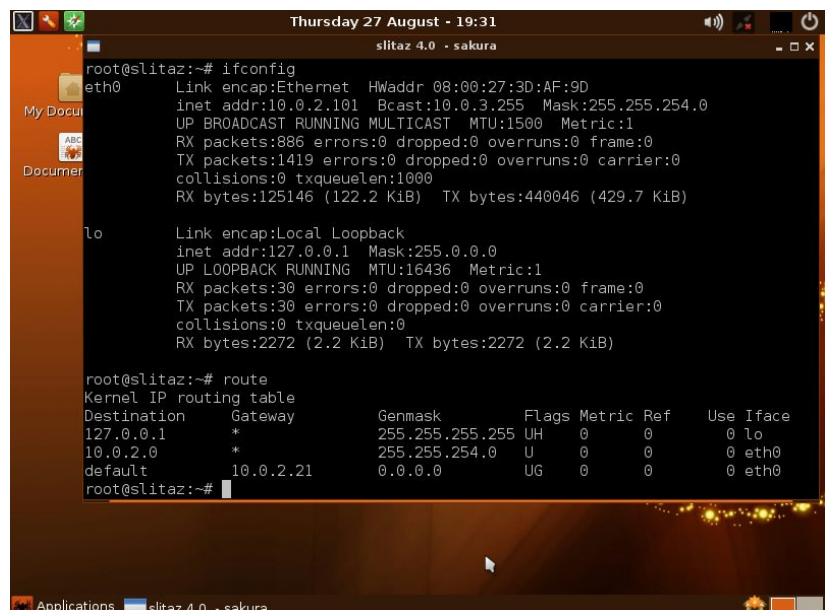


Figura 6: Winbox do roteador 10.0.1.26

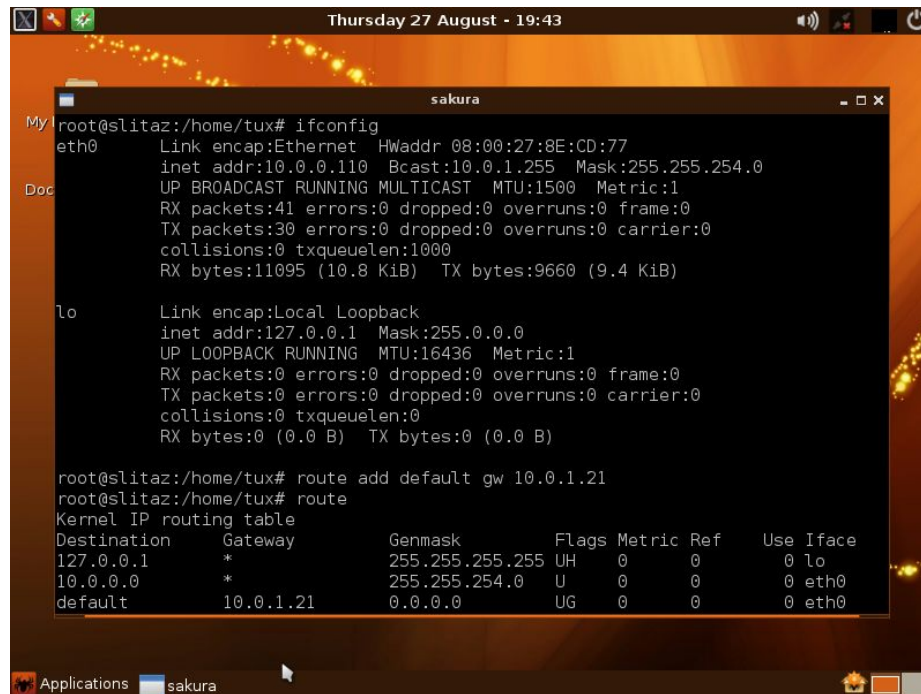
- 3) De acordo com a Figura 11, existem dois domínios de *broadcast*, um para cada sub-rede. Para a sub-rede 10.0.2.0/23, a máscara de rede tem os 23 primeiros bits como 1 e os 9 restantes como 0. Portanto, o primeiro endereço utilizável é 10.0.2.0 até o 10.0.3.255, sendo este último o endereço de *broadcast*. Para a sub-rede 10.0.0.0/23, a faixa vai do 10.0.0.0 até o 10.0.1.255, sendo este último o endereço de *broadcast* igualmente.

A configuração do host 101 é representada pela imagem 7, ao lado. O resultado do comando *route add default* configura a rota padrão para o envio de pacotes, isto é, quando a máscara de rede, que possui nenhum bit diferente de 0, é aplicada ao endereço, o resultado consiste também em um endereço completamente nulo (0.0.0.0) e o pacote é, assim, transmitido ao roteador 10.0.2.21.





Da mesma forma, configura-se o *host* 110, obtendo-se:



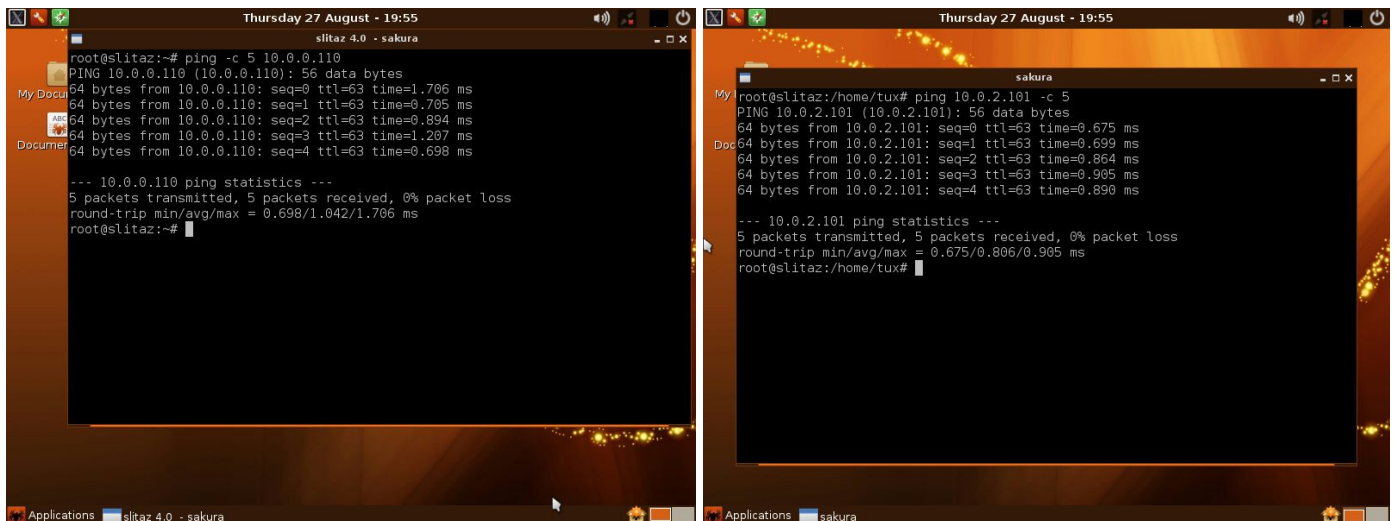
```
My root@slitaz:/home/tux# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:8E:CD:77
          inet addr:10.0.0.110  Bcast:10.0.1.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:41 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11095 (10.8 KiB)  TX bytes:9660 (9.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@slitaz:/home/tux# route add default gw 10.0.1.21
root@slitaz:/home/tux# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0:0:0:0:0:0:0:0  *               *               UH        0      0        0 lo
10.0.0.0         *               255.255.254.0   U         0      0        0 eth0
default          10.0.1.21       0.0.0.0         UG        0      0        0 eth0
```

Figura 8: Configuração do host 10.0.0.110

As figuras mostradas a seguir representam os comandos *ping* executados primeiramente no host 10.0.2.101 (figura 9a) e, em seguida, na máquina 10.0.0.110 (figura 9b).



```
root@slitaz:~# ping -c 5 10.0.0.110
PING 10.0.0.110 (10.0.0.110): 56 data bytes
64 bytes from 10.0.0.110: seq=0 ttl=63 time=1.706 ms
64 bytes from 10.0.0.110: seq=1 ttl=63 time=0.705 ms
64 bytes from 10.0.0.110: seq=2 ttl=63 time=0.894 ms
64 bytes from 10.0.0.110: seq=3 ttl=63 time=1.207 ms
64 bytes from 10.0.0.110: seq=4 ttl=63 time=0.698 ms

--- 10.0.0.110 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.698/1.042/1.706 ms
root@slitaz:~#
```

```
My root@slitaz:/home/tux# ping 10.0.2.101 -c 5
PING 10.0.2.101 (10.0.2.101): 56 data bytes
64 bytes from 10.0.2.101: seq=0 ttl=63 time=0.675 ms
64 bytes from 10.0.2.101: seq=1 ttl=63 time=0.699 ms
64 bytes from 10.0.2.101: seq=2 ttl=63 time=0.864 ms
64 bytes from 10.0.2.101: seq=3 ttl=63 time=0.905 ms
64 bytes from 10.0.2.101: seq=4 ttl=63 time=0.890 ms

--- 10.0.2.101 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.675/0.806/0.905 ms
root@slitaz:/home/tux#
```

Figura 9: (a) ping no host 10.0.2.101 e (b) 10.0.0.110.

- 4) Em relação ao atributo TTL, os pacotes de ambos os comandos valem 63. Considerando que um ping para o mesmo *host* de destino possui pacotes com TTL valendo 64, concluímos, conforme esperado, que os pacotes enviados de 10.0.2.101 a 10.0.0.110 e vice-versa passam por um único roteador, responsável pelo decremento deste parâmetro.

Para acessar a máquina virtual XP, é necessário modificar o *Gateway* padrão assim como foi realizado para as máquinas Linux. Para tal, acessamos o painel de controle e modificamos este endereço para 10.0.1.21, que é o endereço da interface *eth5* do roteador 1. Desta maneira, conseguimos acessar esta máquina a partir de 10.0.2.101. O tráfego gerado pelo comando *ping* é então monitorado pelo *Wireshark* no host Windows, e o resultado está representado na figura 10 a seguir. Foram utilizados os *filters* ICMP e ARP.

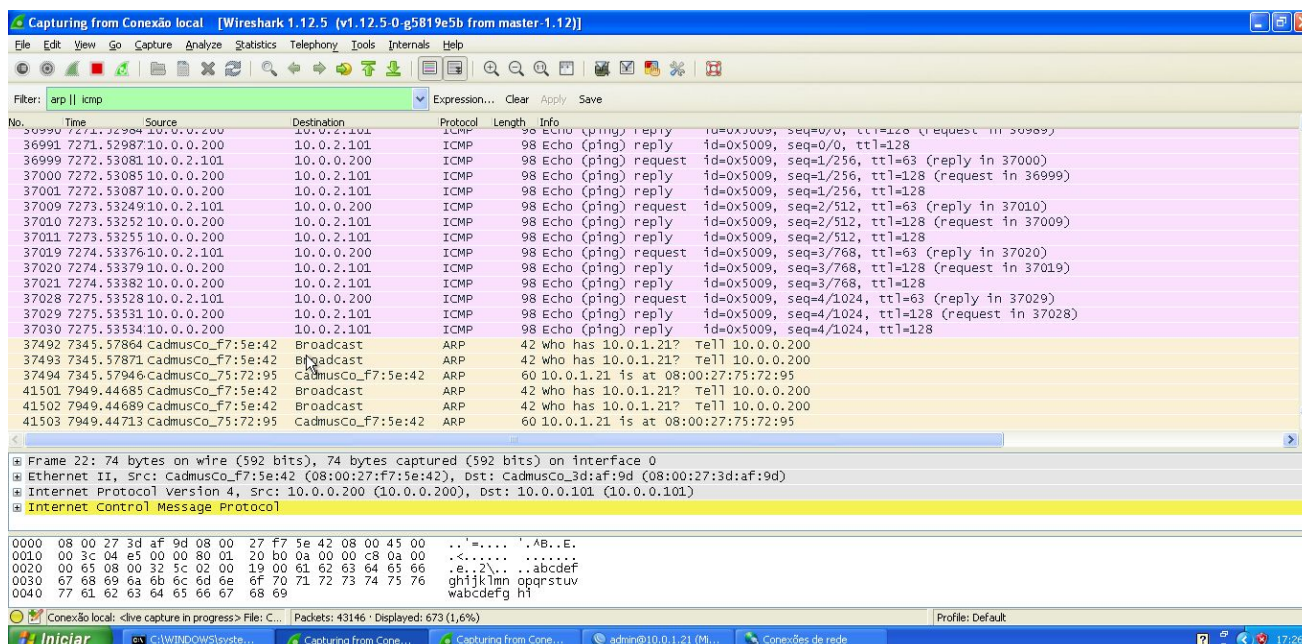


Figura 10: Captura do tráfego dos pacotes gerados por `ping -c 5 10.0.0.200`

Conforme esperado, os pacotes recebidos por `10.0.0.200` possuem TTL já decrementado, já que passaram por um roteador, enquanto os que serão enviados tem este atributo setado como 128. Evidentemente, tais pacotes não passaram por nenhum roteador.

Destacam-se também as mensagens ARP trocadas entre o roteador e a máquina Windows XP, que servem para atualizar a tabela ARP desta última com o endereço MAC deste *router* e as mensagens ICMP, utilizadas pelo comando *ping*.

- 5) Os endereços utilizados para a configuração das rotas estáticas estão esquematizados na figura 11, logo a seguir. Para este exercício, foi adotada a convenção de que os pacotes com destino à subrede `10.0.12.0/23` e origem `10.0.8.0/23`, e vice-versa, adotem o caminho `R1 <-> R5 <-> R4`. Os demais pacotes cuja origem é `10.0.8.0/23` e destino qualquer passam por `R1 <-> R2 <-> R3 <-> R4 <-> R6`.

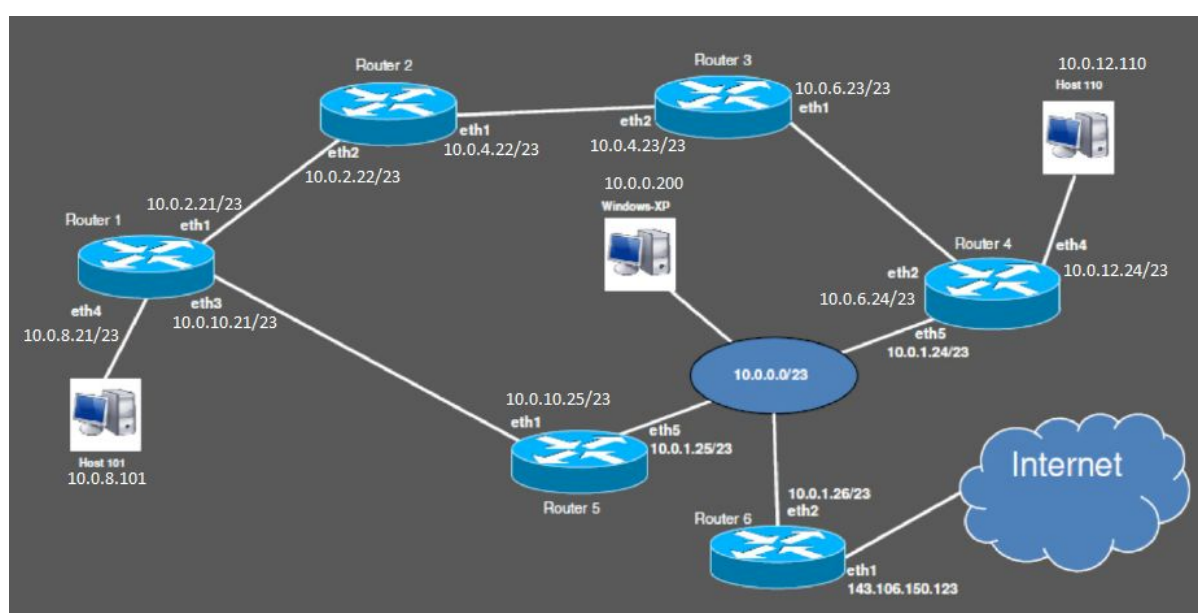


Figura 11: Rede e endereços virtualizados

A tabela de roteamento e os estados das interface do *Router 1* estão representados a seguir:

```
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
#   NAME      TYPE      ACTUAL-MTU  L2MTU
0   R ether1   ether      1500
1   X ether2   ether      1500
2   R ether3   ether      1500
3   R ether4   ether      1500
4   X ether5   ether      1500
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#   DST-ADDRESS  PREF-SRC  GATEWAY      DISTANCE
0   A S 0.0.0.0/0    10.0.2.22  1
1   ADC 10.0.2.0/23   10.0.2.21  0
2   ADC 10.0.8.0/23   10.0.8.21  0
3   ADC 10.0.10.0/23  10.0.10.21 0
4   A S 10.0.12.0/23 10.0.10.25 1
[admin@MikroTik] >
```

Figura 12: Informações do roteador 1.

Observa-se que somente 3 das 5 possíveis interfaces estão ativadas, de acordo com a figura 11. No que se refere à tabela de roteamento, destacam-se duas rotas, as caracterizadas pelas linhas 0 e 4. A primeira,  $0.0.0.0/0 > 10.0.2.22$ , trata-se da rota default do roteador: todos os pacotes que não vão à rede  $10.0.12.0/23$  são enviados para o gateway  $10.0.2.22$  e, conseqüentemente, para o *Router 2*. A outra rota,  $10.0.12.0/23 > 10.0.10.25$ , transporta os pacotes cujo destino é a subrede  $10.0.12.0/23$  para o *Router 5*.

A tabela do roteador 2 é

```
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
#   NAME      TYPE      ACTUAL-MTU  L2MTU
0   R ether1   ether      1500
1   R ether2   ether      1500
2   X ether3   ether      1500
3   X ether4   ether      1500
4   X ether5   ether      1500
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#   DST-ADDRESS  PREF-SRC  GATEWAY      DISTANCE
0   A S 0.0.0.0/0    10.0.4.23  1
1   ADC 10.0.2.0/23   10.0.2.22  0
2   ADC 10.0.4.0/23   10.0.4.22  0
3   A S 10.0.8.0/23   10.0.2.21  1
[admin@MikroTik] >
```

Figura 13: Informações do roteador 2.

Pacotes cujo destino é a subrede  $10.0.8.0/23$  são transportados para o *Router 1* via o gateway  $10.0.2.21$  e o restante, para o *Router 3* via o gateway  $10.0.4.23$ .



Para o roteador 3:

```
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
#    NAME      TYPE      ACTUAL-MTU  L2MTU
0    R  ether1    ether      1500
1    R  ether2    ether      1500
2    X  ether3    ether      1500
3    X  ether4    ether      1500
4    X  ether5    ether      1500
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#    DST-ADDRESS  PREF-SRC  GATEWAY      DISTANCE
0    A S  0.0.0.0/0      10.0.6.24    1
1    ADC 10.0.4.0/23     10.0.4.23    ether2        0
2    ADC 10.0.6.0/23     10.0.6.23    ether1        0
3    A S  10.0.8.0/23     10.0.4.22    1
```

Figura 14: Informações do roteador 3.

O mesmo princípio se aplica: os pacotes de destino a *10.0.8.0/23* são reenviados ao roteador 2 ou encaminhados ao *Router 4* via a interface 1.

O roteador 4 é descrito por:

```
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
#    NAME      TYPE      ACTUAL-MTU  L2MTU
0    X  ether1    ether      1500
1    R  ether2    ether      1500
2    X  ether3    ether      1500
3    R  ether4    ether      1500
4    R  ether5    ether      1500
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#    DST-ADDRESS  PREF-SRC  GATEWAY      DISTANCE
0    A S  0.0.0.0/0      10.0.1.26    1
1    ADC 10.0.0.0/23     10.0.1.24    ether5        0
2    ADC 10.0.6.0/23     10.0.6.24    ether2        0
3    A S  10.0.8.0/23     10.0.1.25    1
4    ADC 10.0.12.0/23    10.0.12.24    ether4        0
```

Figura 15: Informações do roteador 4.

Pacotes com destino à subrede *10.0.8.0/23* são enviados ao *gateway 10.0.1.25* e, consequentemente, para o *Router 5*. Essa rota consiste na primeira parte do caminho de volta para esta subrede, no caso de um comando ping para o host *10.0.12.110*. O roteador 4 recebe a resposta deste *host* e consulta sua tabela de roteamento. Verificando que o seu destinatário está na rede *10.0.8.0/23*, ele encaminha tal pacote para o respectivo *gateway*.

Para o roteador 5:

```
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
#    NAME    TYPE    ACTUAL-MTU  L2MTU
0    R ether1    ether      1500
1    X ether2    ether      1500
2    X ether3    ether      1500
3    X ether4    ether      1500
4    R ether5    ether      1500
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#    DST-ADDRESS    PREF-SRC    GATEWAY    DISTANCE
0    A S 0.0.0.0/0        10.0.1.26      1
1    ADC 10.0.0.0/23      10.0.1.25      0
2    A S 10.0.8.0/23      10.0.10.21     1
3    ADC 10.0.10.0/23     10.0.10.25     0
4    A S 10.0.12.0/23    10.0.1.24      1
[admin@MikroTik] > _
```

Figura 16: Informações do roteador 5.

Aponta-se as rotas  $10.0.8.0/23 > 10.0.10.21$  (Router 1) e  $10.0.12.0/23 > 10.0.1.24$  (Router 5), responsáveis por encaminhar os pacotes entre as duas subredes dos hosts. A comunicação entre eles é portanto habilitada a partir da configuração deste roteador.

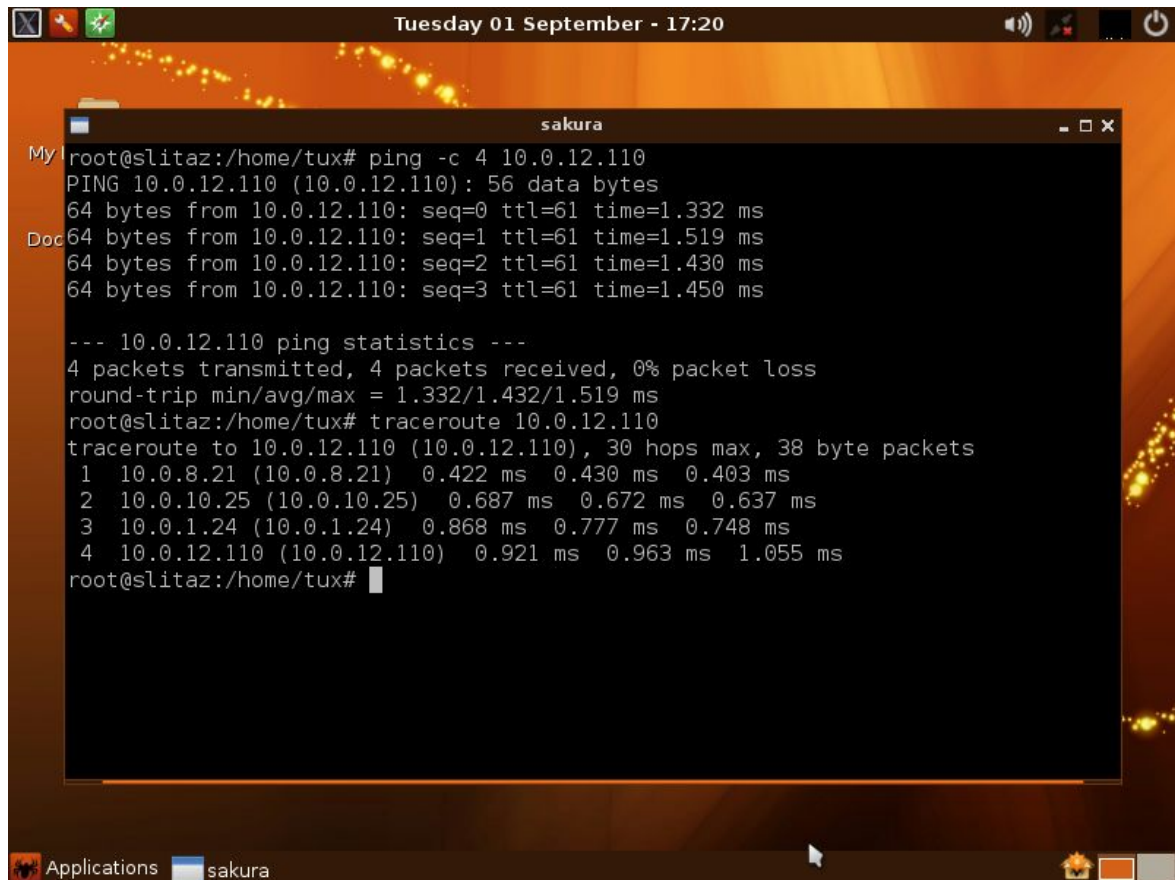
Finalmente, para o roteador 6:

```
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
#    NAME    TYPE    ACTUAL-MTU  L2MTU
0    R ether1    ether      1500
1    R ether2    ether      1500
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#    DST-ADDRESS    PREF-SRC    GATEWAY    DISTANCE
0    S 0.0.0.0/0        143.106.150.123 1
1    A S 0.0.0.0/0        10.0.2.2        1
2    ADC 10.0.0.0/23      10.0.1.26      0
3    ADC 10.0.2.0/24      10.0.2.15      0
4    A S 10.0.8.0/23      10.0.1.24      1
5    A S 10.0.12.0/23     10.0.1.24      1
[admin@MikroTik] > _
```

Figura 17: Informações do roteador 6



O resultado dos comandos *ping* e *traceroute* à partir do host *10.0.8.101* para *10.0.12.110* nos retorna:



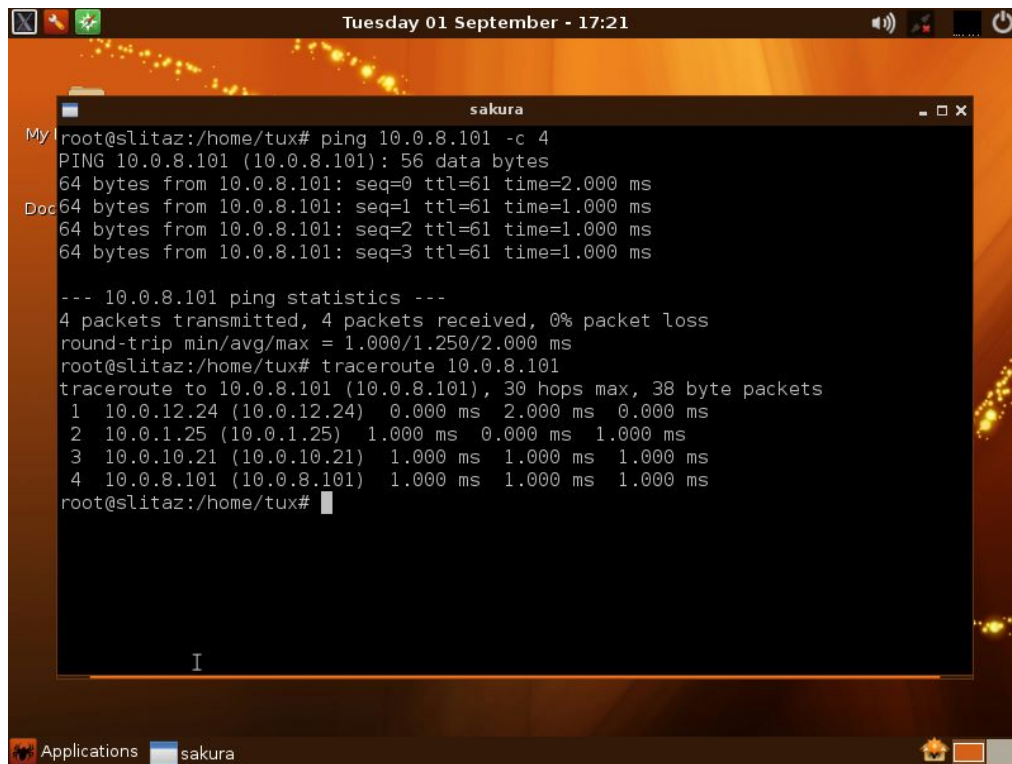
```
root@slitaz:/home/tux# ping -c 4 10.0.12.110
PING 10.0.12.110 (10.0.12.110): 56 data bytes
64 bytes from 10.0.12.110: seq=0 ttl=61 time=1.332 ms
64 bytes from 10.0.12.110: seq=1 ttl=61 time=1.519 ms
64 bytes from 10.0.12.110: seq=2 ttl=61 time=1.430 ms
64 bytes from 10.0.12.110: seq=3 ttl=61 time=1.450 ms

--- 10.0.12.110 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.332/1.432/1.519 ms
root@slitaz:/home/tux# traceroute 10.0.12.110
traceroute to 10.0.12.110 (10.0.12.110), 30 hops max, 38 byte packets
 1  10.0.8.21 (10.0.8.21)  0.422 ms  0.430 ms  0.403 ms
 2  10.0.10.25 (10.0.10.25)  0.687 ms  0.672 ms  0.637 ms
 3  10.0.1.24 (10.0.1.24)  0.868 ms  0.777 ms  0.748 ms
 4  10.0.12.110 (10.0.12.110)  0.921 ms  0.963 ms  1.055 ms
root@slitaz:/home/tux#
```

Figura 18: Ping e traceroute para 10.0.12.110

Em relação ao comando *ping*, destaca-se o valor do atributo **TTL**, que vale 61. Isso significa que os pacotes sofreram 3 decrementações durante seu percurso. Tal afirmação é confirmada pelo comando *traceroute*, cuja saída é o conjunto de roteadores e hosts presentes no caminho. Conclui-se portanto que os pacotes deixam o *host* *10.0.8.101* por seu gateway padrão, isto é, *10.0.8.21*, passam pelo roteador 1, que, após consultar sua tabela de roteamento, os encaminha ao roteador 5 via o *gateway* *10.0.10.25*. Em seguida, este roteador faz o mesmo e os envia para o roteador 4, pelo *gateway* *10.0.1.24*. Enfim, este último entrega as requisições a *10.0.12.110*, que as responde fazendo o mesmo caminho, mas na direção inversa.

Analogamente, o resultado dos comandos *ping* e *traceroute* à partir do host *10.0.12.110* para *10.0.8.101* nos retorna:



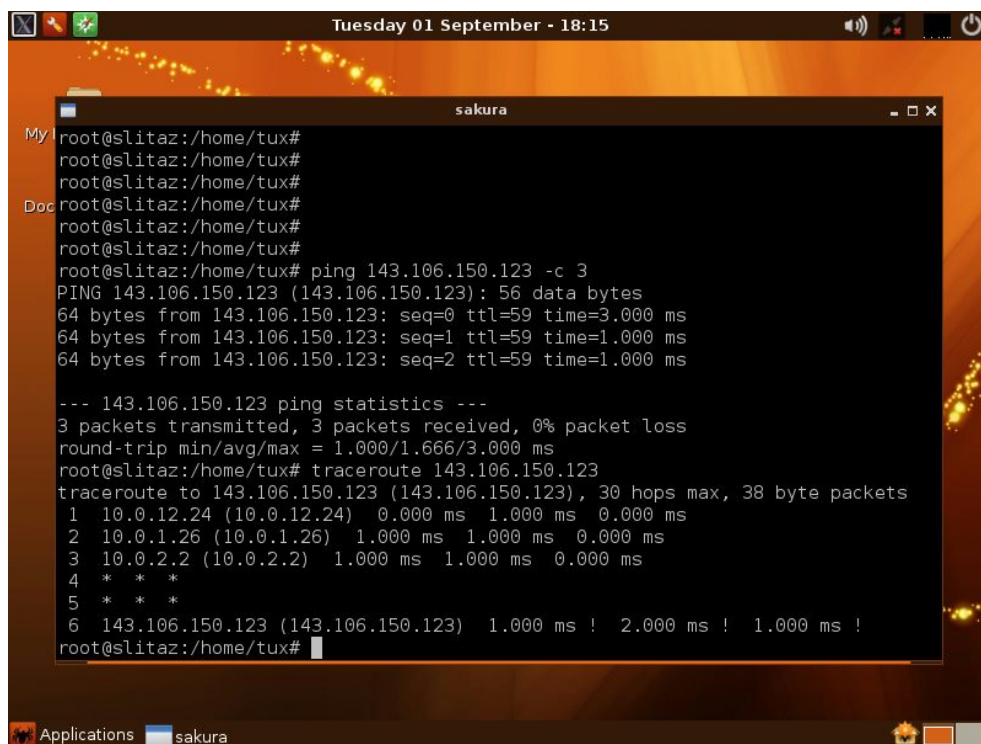
```
My root@slitaz:/home/tux# ping 10.0.8.101 -c 4
PING 10.0.8.101 (10.0.8.101): 56 data bytes
64 bytes from 10.0.8.101: seq=0 ttl=61 time=2.000 ms
64 bytes from 10.0.8.101: seq=1 ttl=61 time=1.000 ms
64 bytes from 10.0.8.101: seq=2 ttl=61 time=1.000 ms
64 bytes from 10.0.8.101: seq=3 ttl=61 time=1.000 ms

--- 10.0.8.101 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.000/1.250/2.000 ms
root@slitaz:/home/tux# traceroute 10.0.8.101
traceroute to 10.0.8.101 (10.0.8.101), 30 hops max, 38 byte packets
 1  10.0.12.24 (10.0.12.24)  0.000 ms  2.000 ms  0.000 ms
 2  10.0.1.25 (10.0.1.25)  1.000 ms  0.000 ms  1.000 ms
 3  10.0.10.21 (10.0.10.21)  1.000 ms  1.000 ms  1.000 ms
 4  10.0.8.101 (10.0.8.101)  1.000 ms  1.000 ms  1.000 ms
root@slitaz:/home/tux#
```

Figura 19: Ping e traceroute para 10.0.8.101

O valor de **TTL** é evidentemente o mesmo, já que os pacotes retornam pelo mesmo percurso e passam pelo mesmo número de roteadores. Em relação à rota, os pacotes saem do *host* 10.0.12.110, passam pelo *router* 4 e são enviados ao roteador 5, via o *gateway* 10.0.1.25. Este último os envia para o roteador 1, via 10.0.10.21, que os entrega a 10.0.8.101 finalmente.

Finalmente, para o comando *ping* à partir de ambos os hosts, obtemos as duas imagens seguintes:

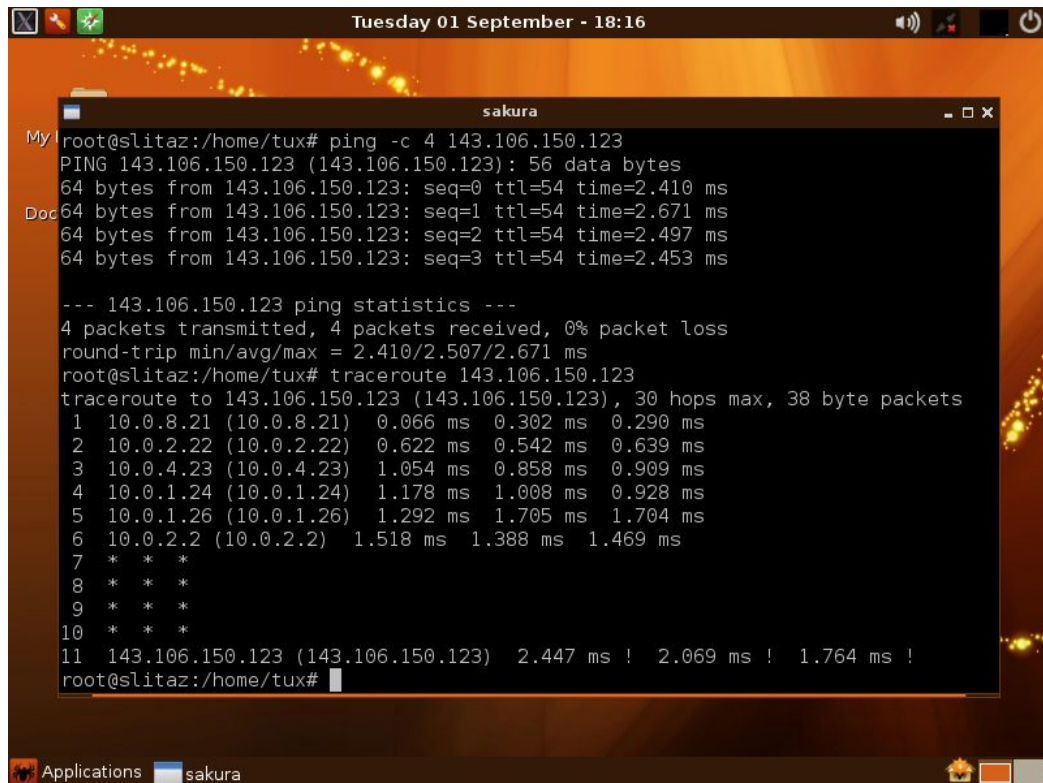


```
My root@slitaz:/home/tux#
root@slitaz:/home/tux#
root@slitaz:/home/tux#
root@slitaz:/home/tux#
root@slitaz:/home/tux#
root@slitaz:/home/tux#
root@slitaz:/home/tux# ping 143.106.150.123 -c 3
PING 143.106.150.123 (143.106.150.123): 56 data bytes
64 bytes from 143.106.150.123: seq=0 ttl=59 time=3.000 ms
64 bytes from 143.106.150.123: seq=1 ttl=59 time=1.000 ms
64 bytes from 143.106.150.123: seq=2 ttl=59 time=1.000 ms

--- 143.106.150.123 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.000/1.666/3.000 ms
root@slitaz:/home/tux# traceroute 143.106.150.123
traceroute to 143.106.150.123 (143.106.150.123), 30 hops max, 38 byte packets
 1  10.0.12.24 (10.0.12.24)  0.000 ms  1.000 ms  0.000 ms
 2  10.0.1.26 (10.0.1.26)  1.000 ms  1.000 ms  0.000 ms
 3  10.0.2.2 (10.0.2.2)  1.000 ms  1.000 ms  0.000 ms
 4  * * *
 5  * * *
 6  143.106.150.123 (143.106.150.123)  1.000 ms !  2.000 ms !  1.000 ms !
root@slitaz:/home/tux#
```

Figura 19: Ping e traceroute para 143.106.156.123 a partir de 10.0.12.110

Observa-se que os pacotes passam somente por R4, R5 e R6 antes de serem encaminhados para 10.0.2.2. O percurso de retorno dos pacotes gerados por *ping* é o mesmo.



```
Tuesday 01 September - 18:16

sakura

root@slitaz:/home/tux# ping -c 4 143.106.150.123
PING 143.106.150.123 (143.106.150.123): 56 data bytes
64 bytes from 143.106.150.123: seq=0 ttl=54 time=2.410 ms
64 bytes from 143.106.150.123: seq=1 ttl=54 time=2.671 ms
64 bytes from 143.106.150.123: seq=2 ttl=54 time=2.497 ms
64 bytes from 143.106.150.123: seq=3 ttl=54 time=2.453 ms

--- 143.106.150.123 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 2.410/2.507/2.671 ms
root@slitaz:/home/tux# traceroute 143.106.150.123
traceroute to 143.106.150.123 (143.106.150.123), 30 hops max, 38 byte packets
 1  10.0.8.21 (10.0.8.21)  0.066 ms  0.302 ms  0.290 ms
 2  10.0.2.22 (10.0.2.22)  0.622 ms  0.542 ms  0.639 ms
 3  10.0.4.23 (10.0.4.23)  1.054 ms  0.858 ms  0.909 ms
 4  10.0.1.24 (10.0.1.24)  1.178 ms  1.008 ms  0.928 ms
 5  10.0.1.26 (10.0.1.26)  1.292 ms  1.705 ms  1.704 ms
 6  10.0.2.2 (10.0.2.2)  1.518 ms  1.388 ms  1.469 ms
 7  * * *
 8  * * *
 9  * * *
10  * * *
11 143.106.150.123 (143.106.150.123)  2.447 ms !  2.069 ms !  1.764 ms !
root@slitaz:/home/tux#
```

Figura 20: Ping e traceroute para 143.106.156.123 a partir de 10.0.8.101

Por escolha própria, os pacotes gerados por *10.0.8.101* que não serão enviados a *10.0.12.0/23* atravessam os roteadores R2 e R3 ao invés de seguirem por R5. Porém, tais pacotes não seguem esta mesma rota para a volta. Isso porque quando o roteador R6 verifica que ele precisa enviar pacotes cujo destino é esta subrede, ele os transmite a R4, que por sua vez, está configurado para os enviar a R5 e não a R3, conforme explicado no caso do comando ping entre os hosts *10.0.8.101* <-> *10.0.12.110*.