Лабораторная работа №6
Моделирование сетей передачи данных

Исаев Б. А.

2025

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

Запуск лабораторной топологии

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  bced047e0c3bcbb01e84e95f729f6f92
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  bced047e0c3bcbb01e84e95f729f6f9
2
root@mininet-vm:~# logout
mininet@mininet-vm:~$
```

Рис. 1: Исправление прав запуска X-соединения в виртуальной машине mininet

Запуск лабораторной топологии

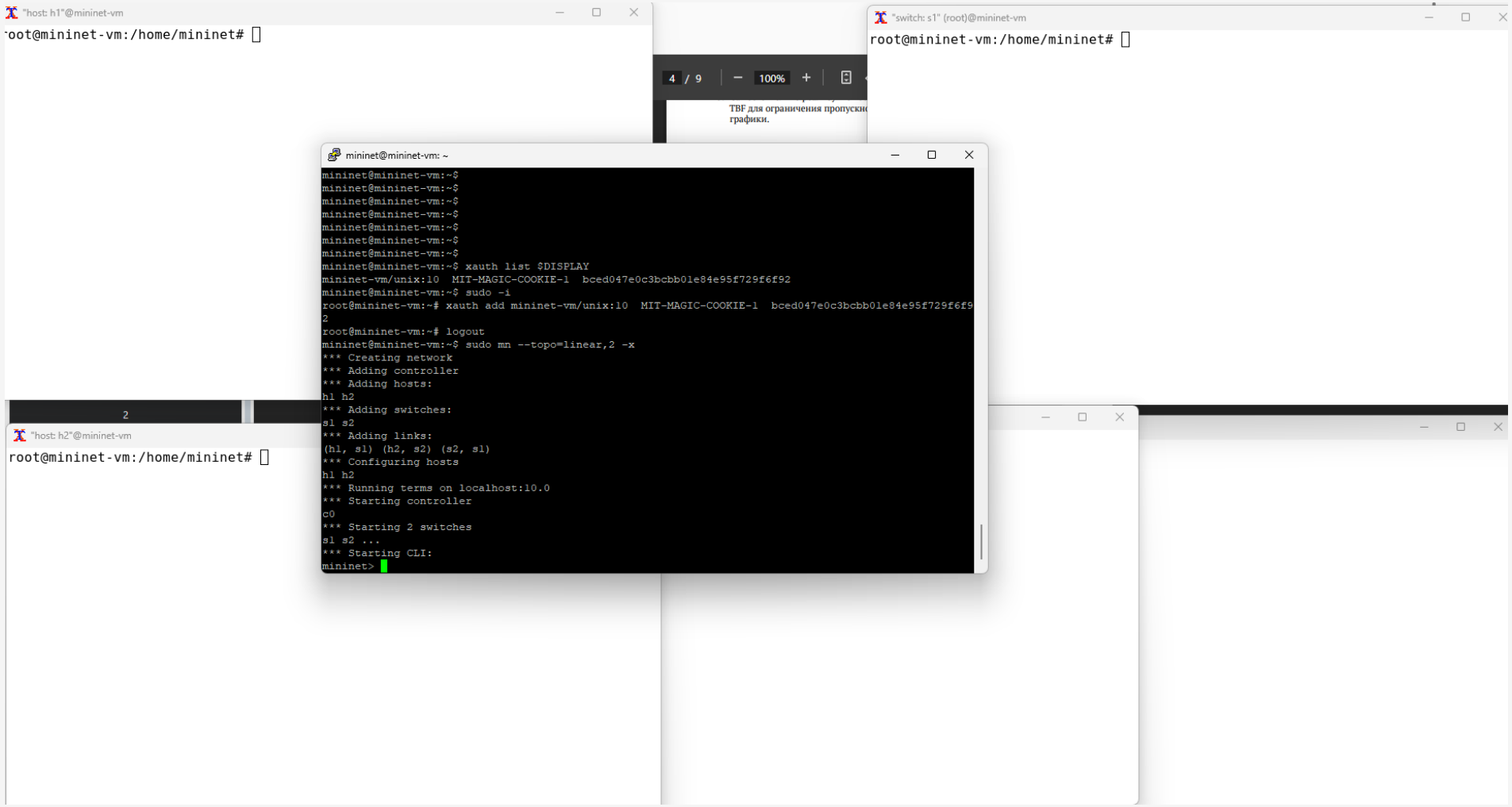

Рис. 2: Создание топологии с двумя хостами и двумя коммутаторами

Запуск лабораторной топологии

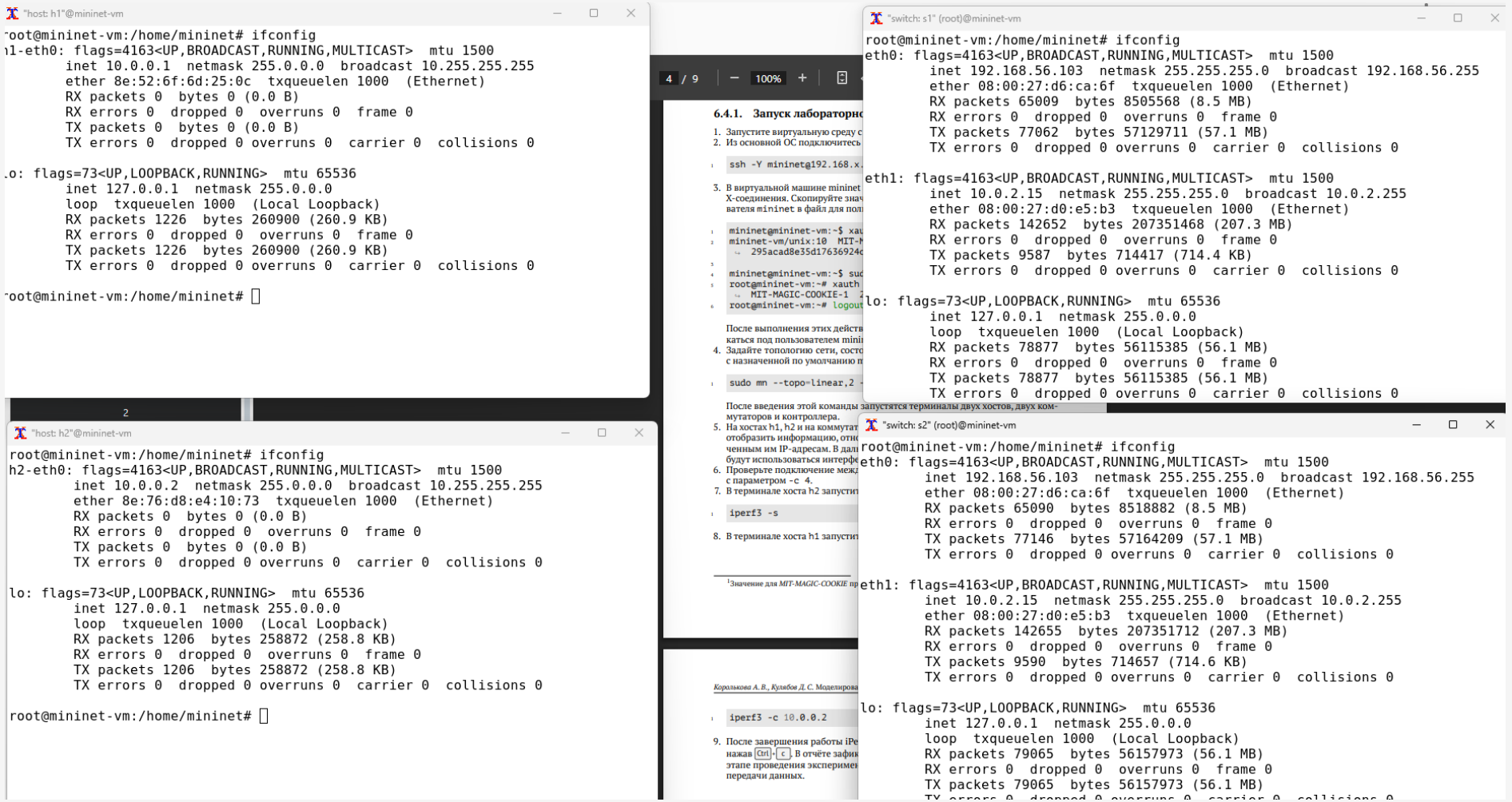

Рис. 3: Отображение информации сетевых интерфейсов и IP-адресов

Запуск лабораторной топологии

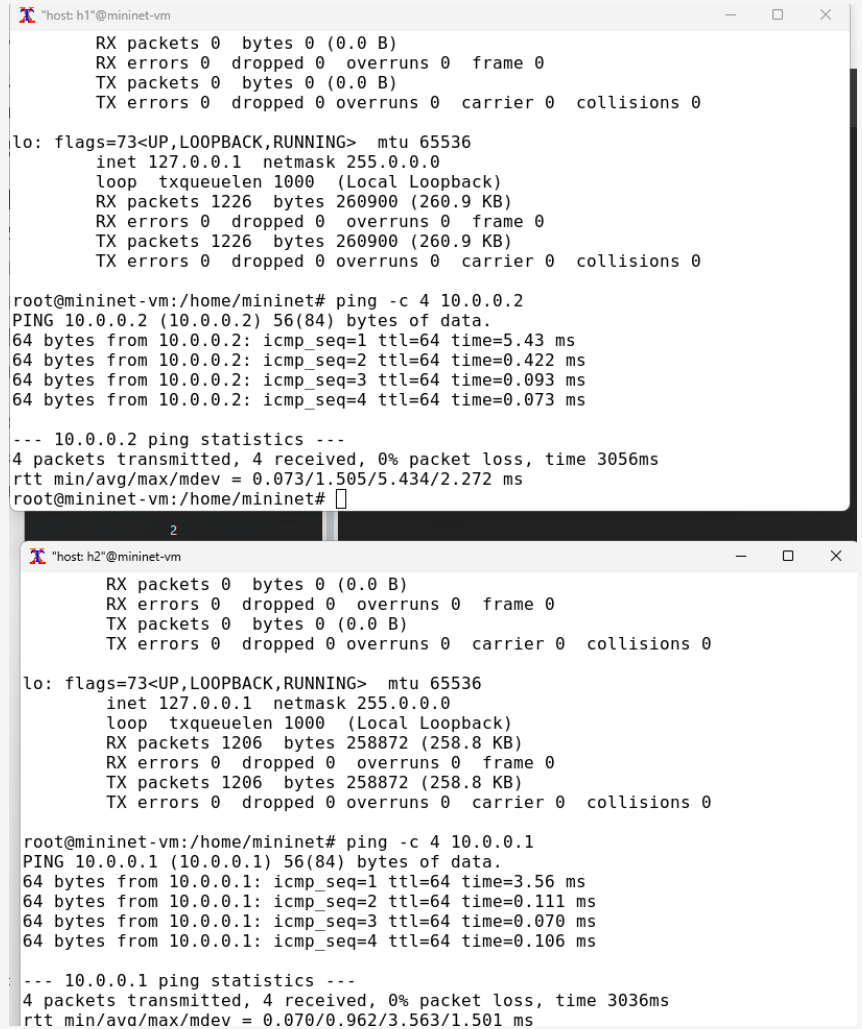

Рис. 4: Проверка подключения между хостами h1 и h2

Запуск лабораторной топологии

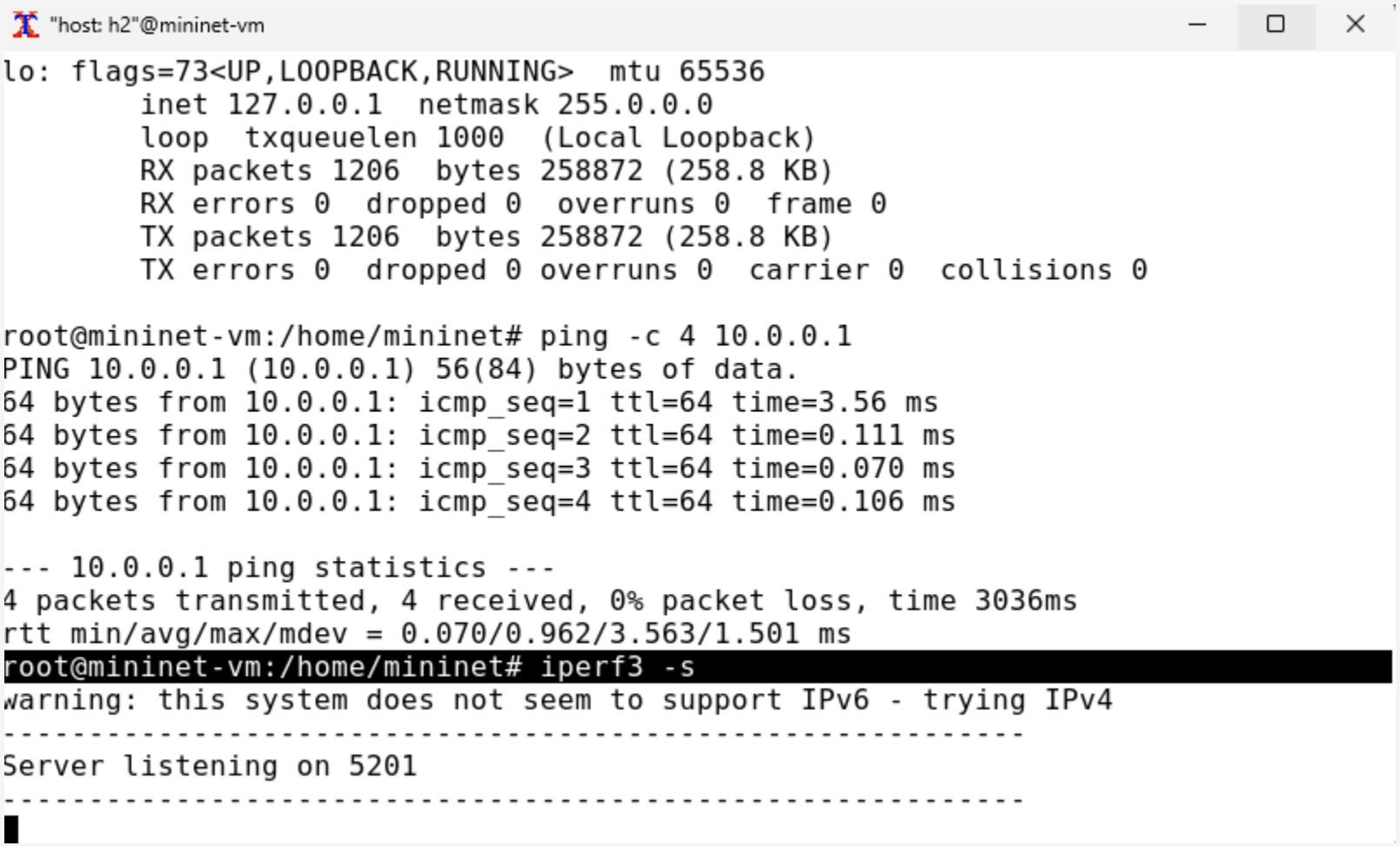

Рис. 5: Запуск iPerf3 в режиме сервера на хосте h2

Запуск лабораторной топологии

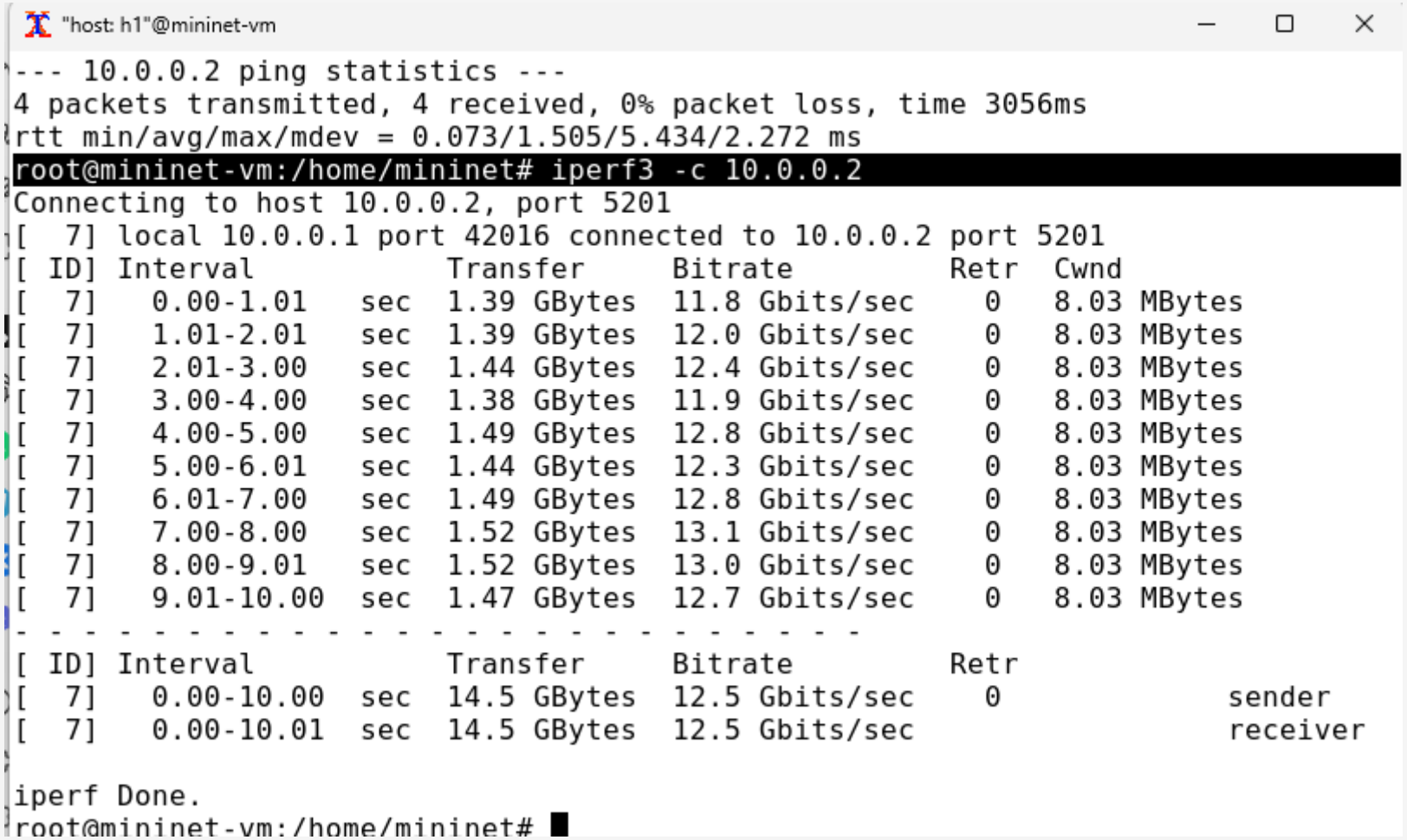

Рис. 6: Запуск iPerf3 в режиме клиента на хосте h1

Запуск лабораторной топологии

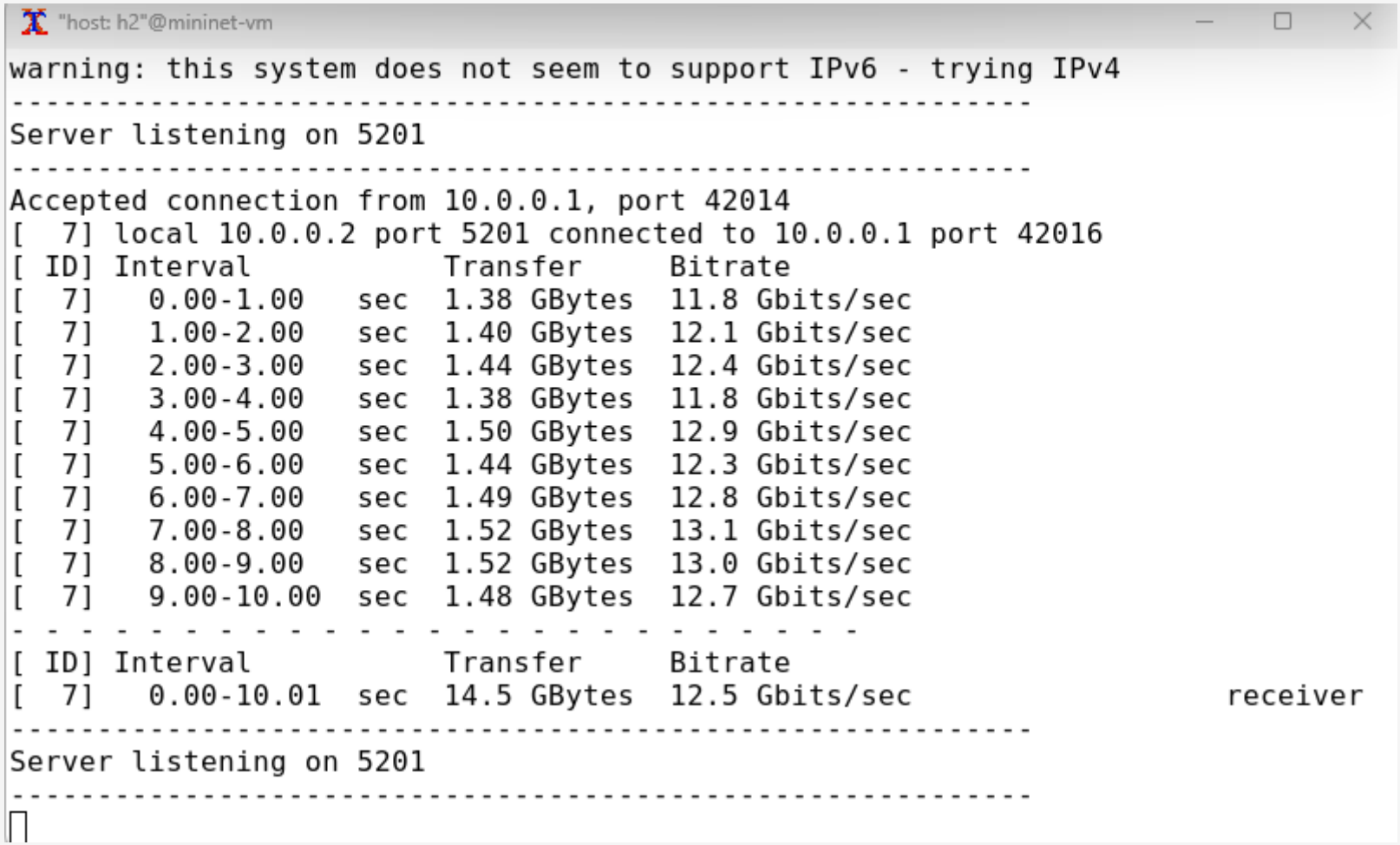

Рис. 7: Остановка iPerf3

Интерактивные эксперименты

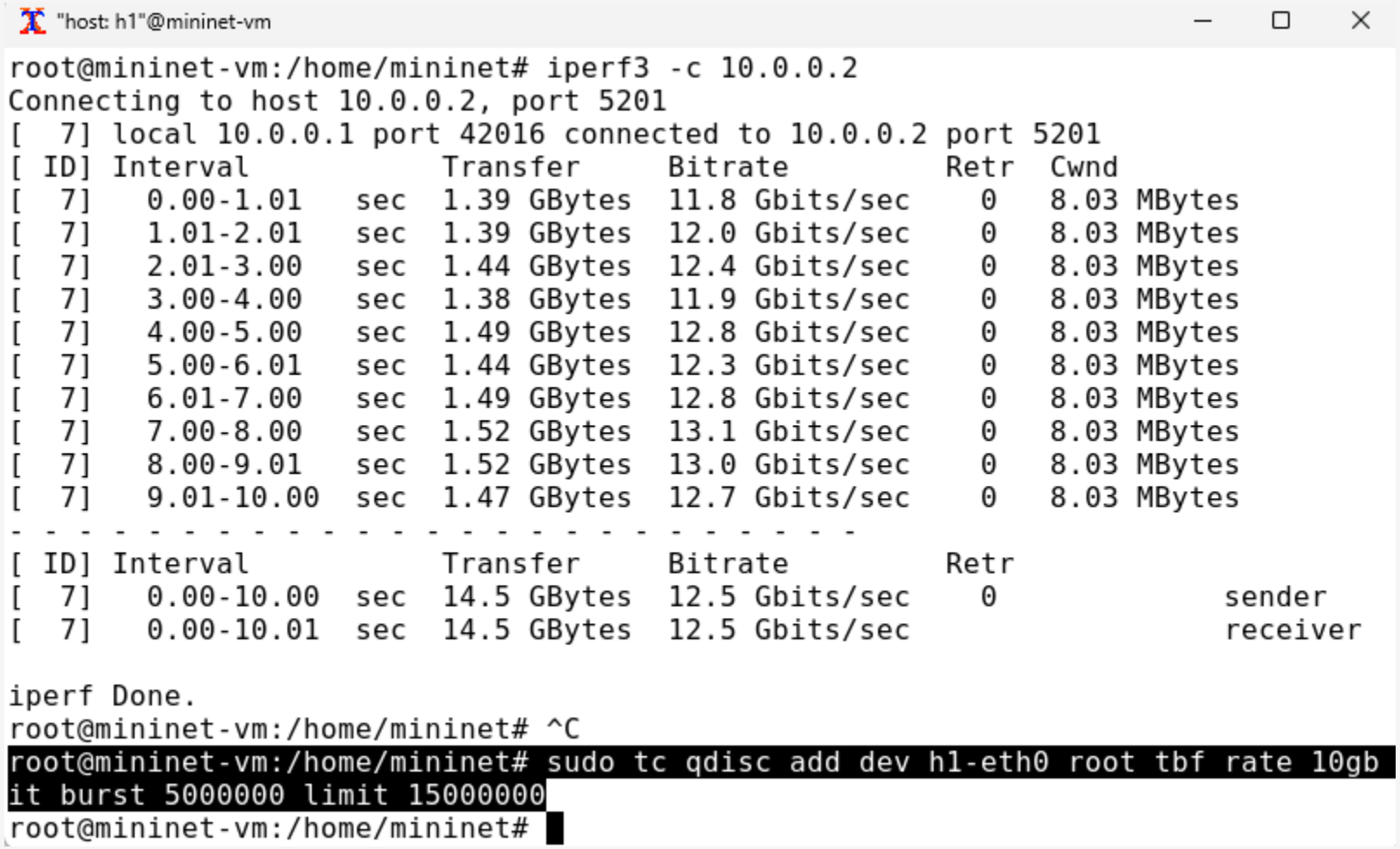

Рис. 8: Изменение пропускной способности хоста h1

Интерактивные эксперименты

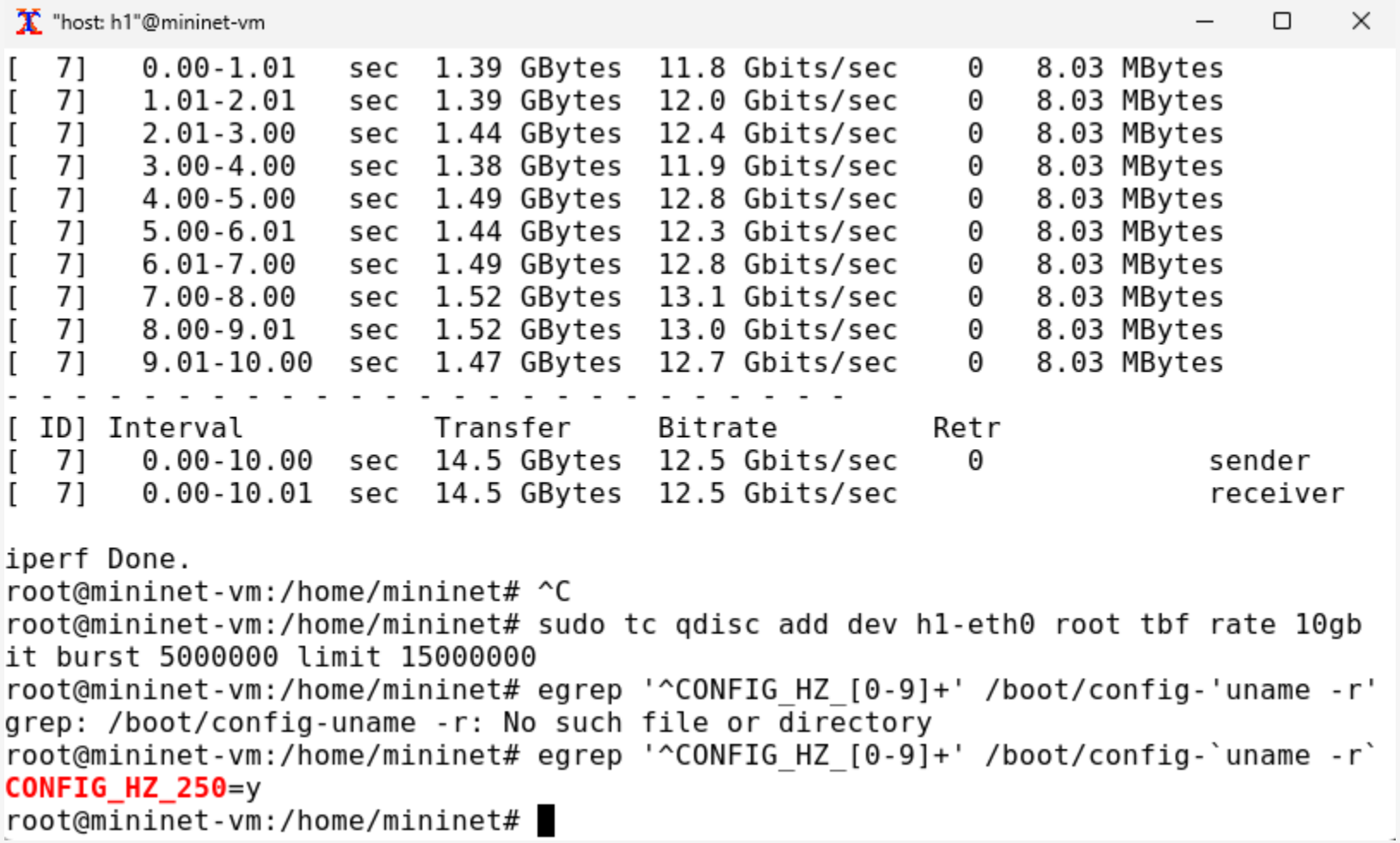

Рис. 9: Установка значения всплеска при ограничении скорости для фильтра tbf

Интерактивные эксперименты

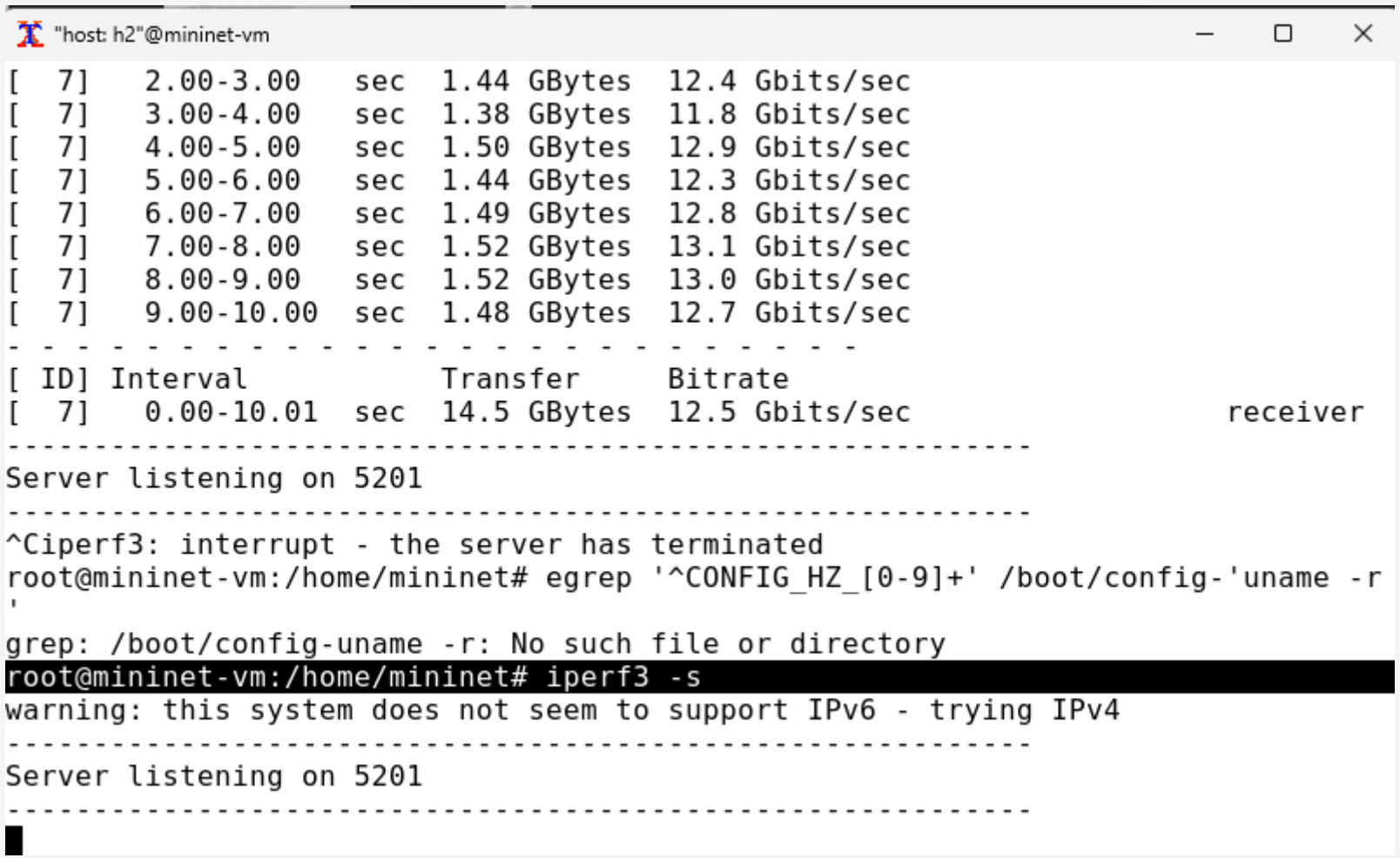

Рис. 10: Запуск iPerf3 в режиме сервера на хосте h2

Интерактивные эксперименты

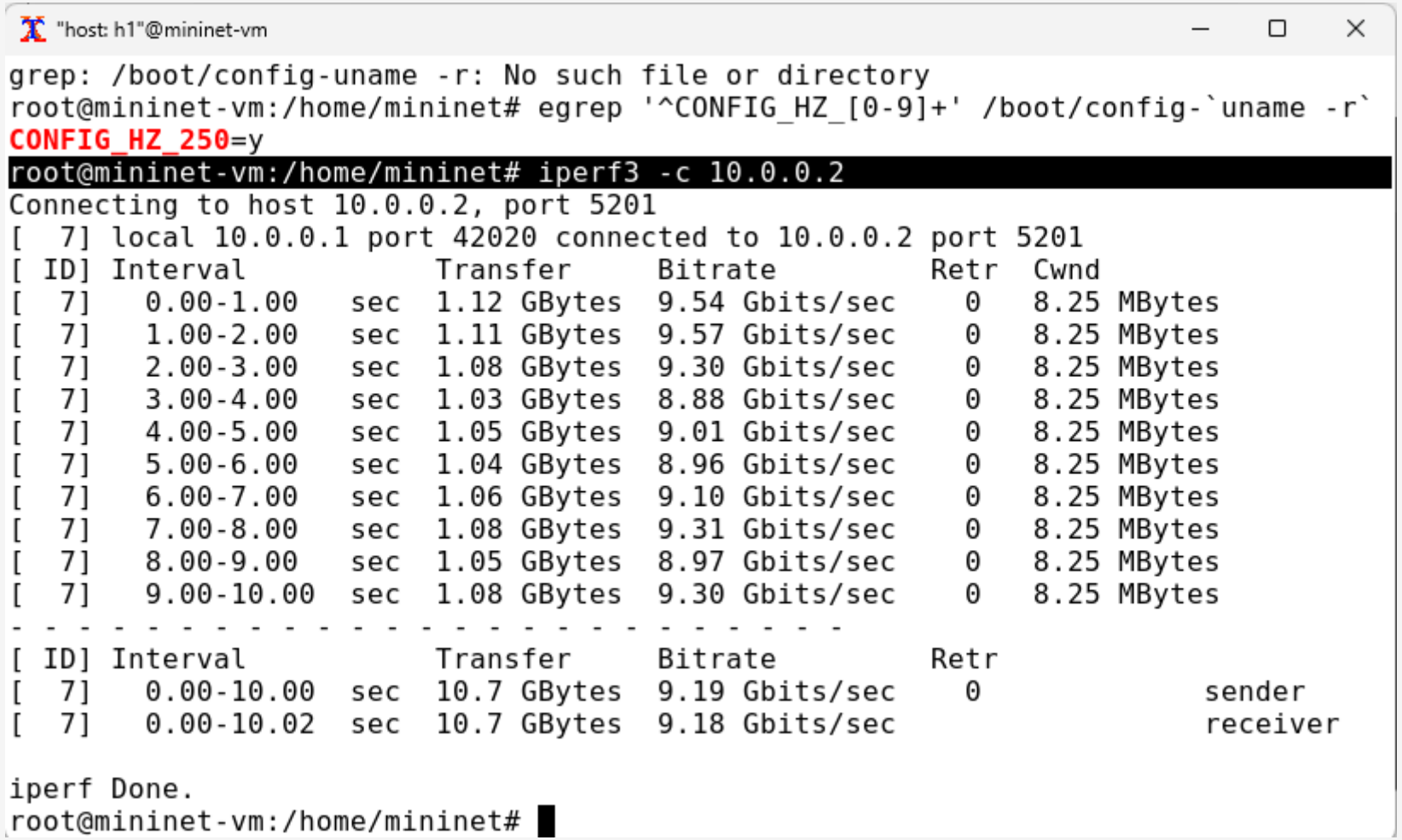

Рис. 11: Запуск iPerf3 в режиме клиента на хосте h1

Интерактивные эксперименты

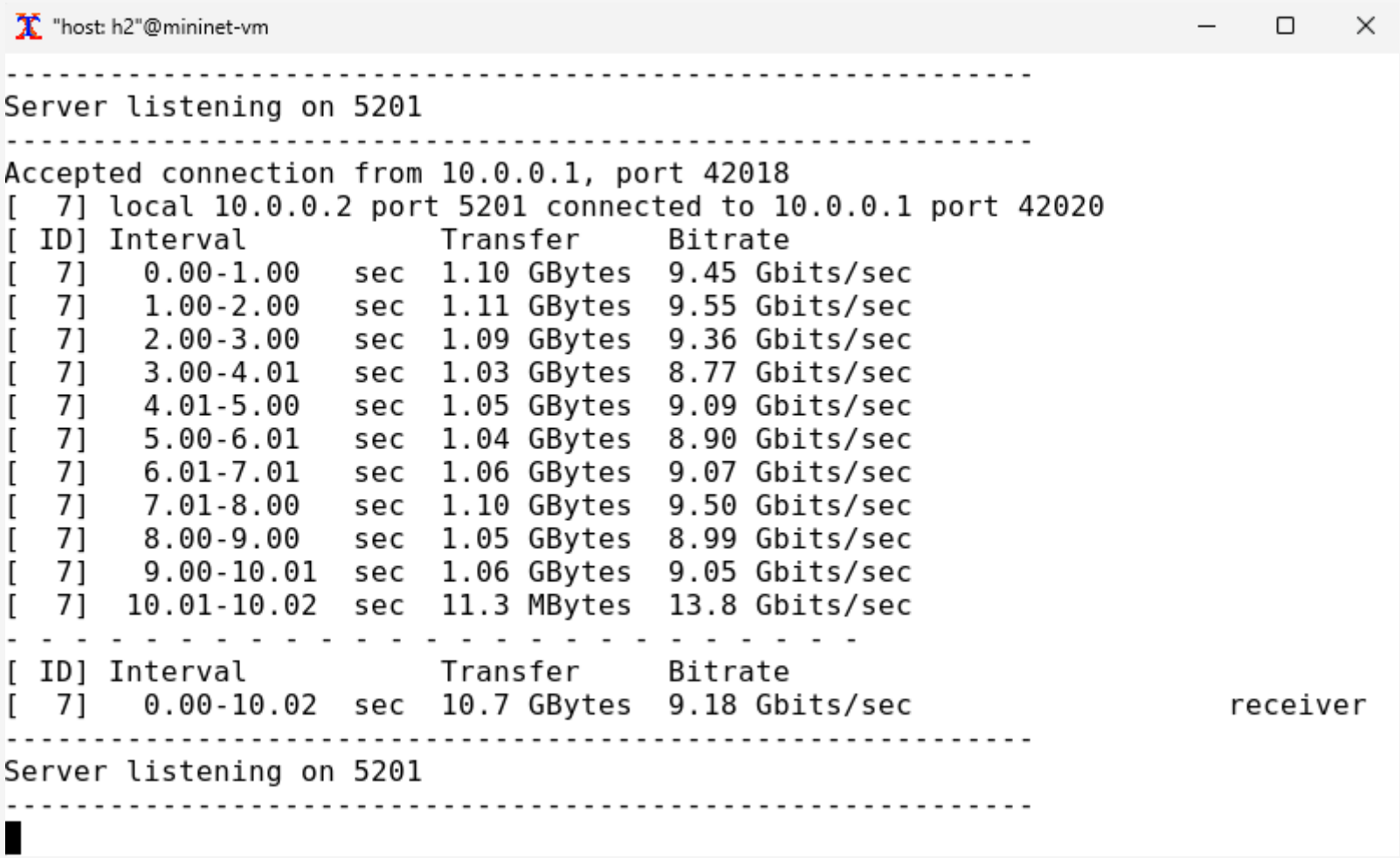

Рис. 12: Остановка iPerf3

Интерактивные эксперименты

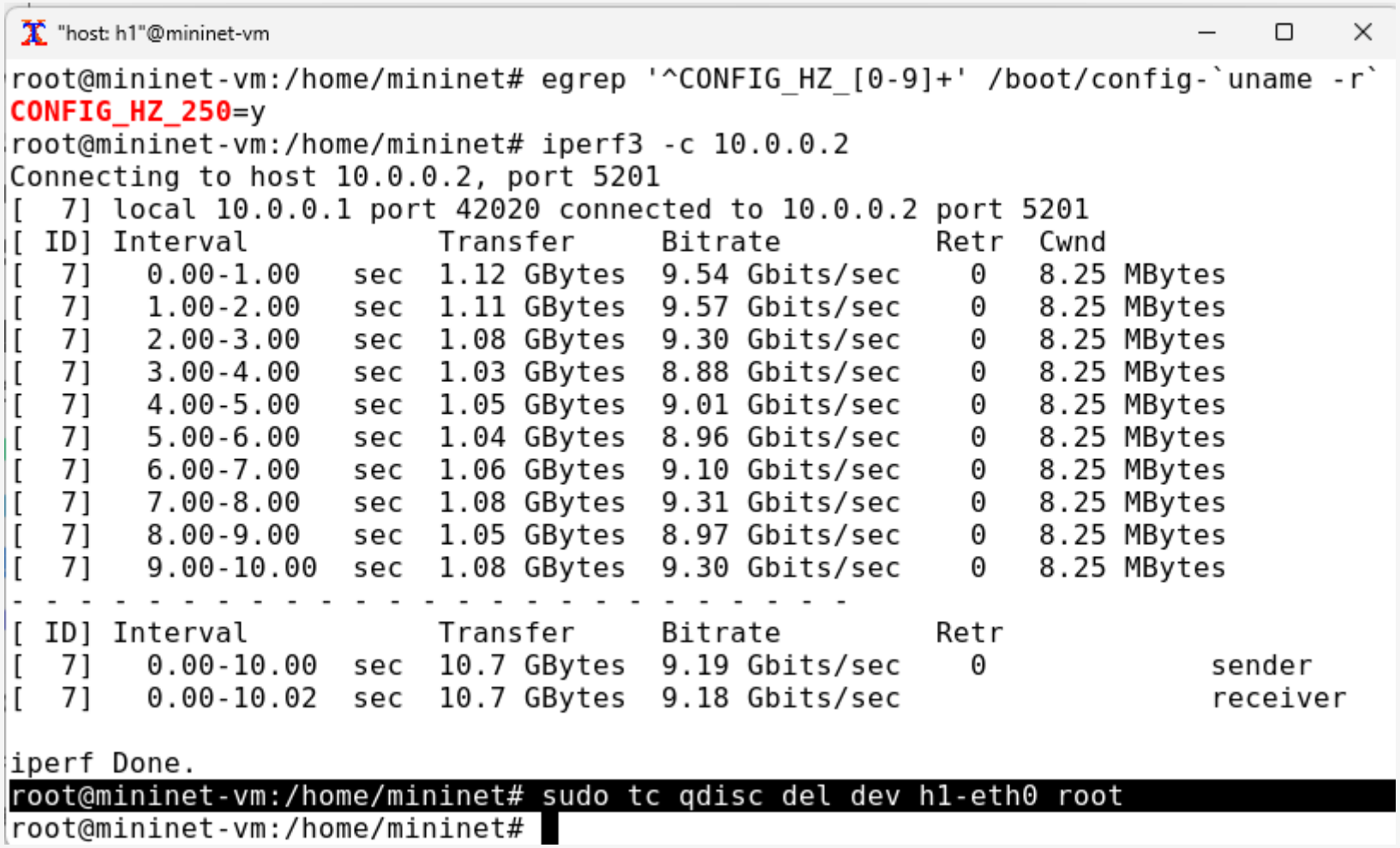

Рис. 13: Удаление модифицированной конфигурации на хосте h1

Интерактивные эксперименты

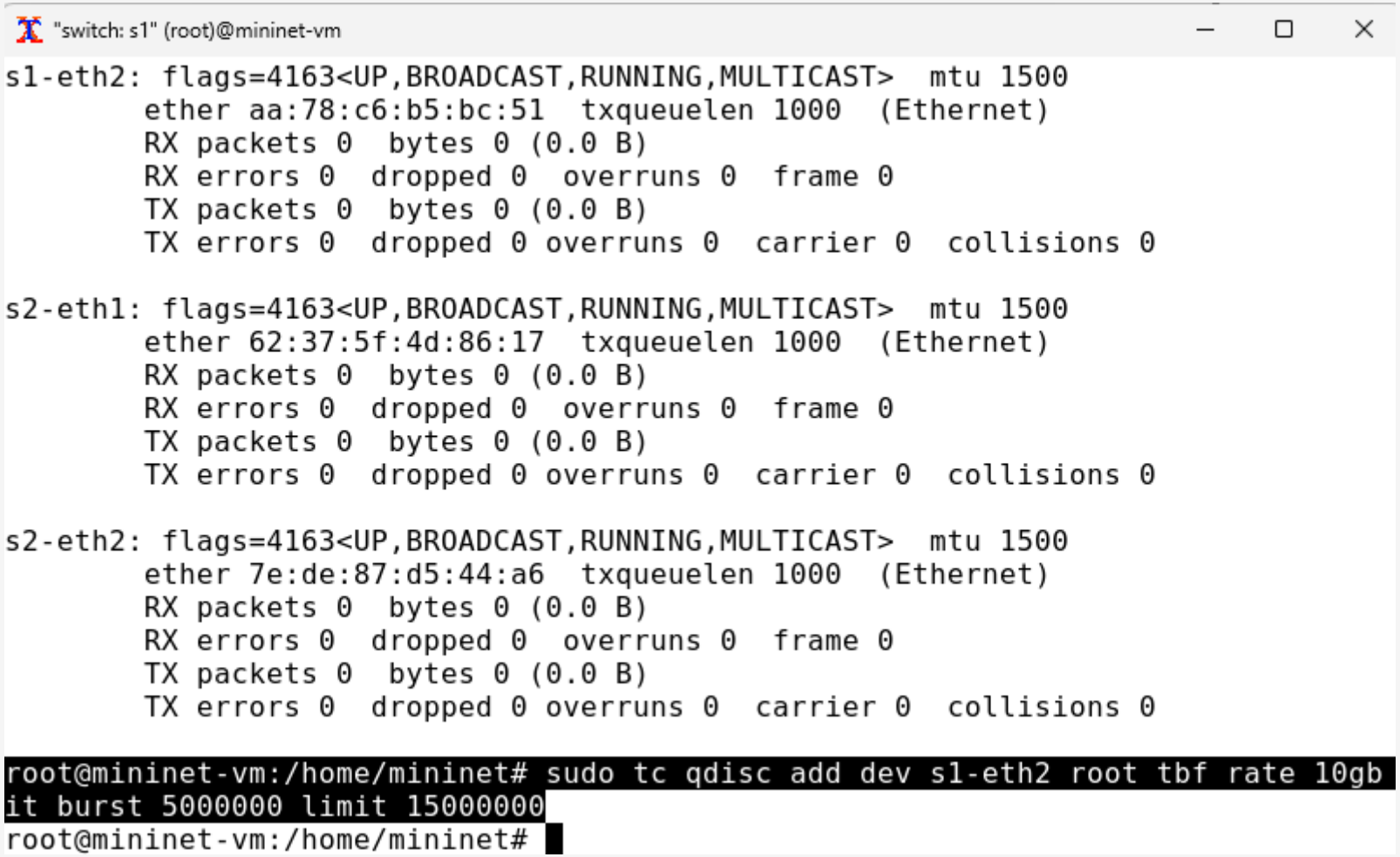

Рис. 14: Применение правила ограничения скорости tbf

Интерактивные эксперименты

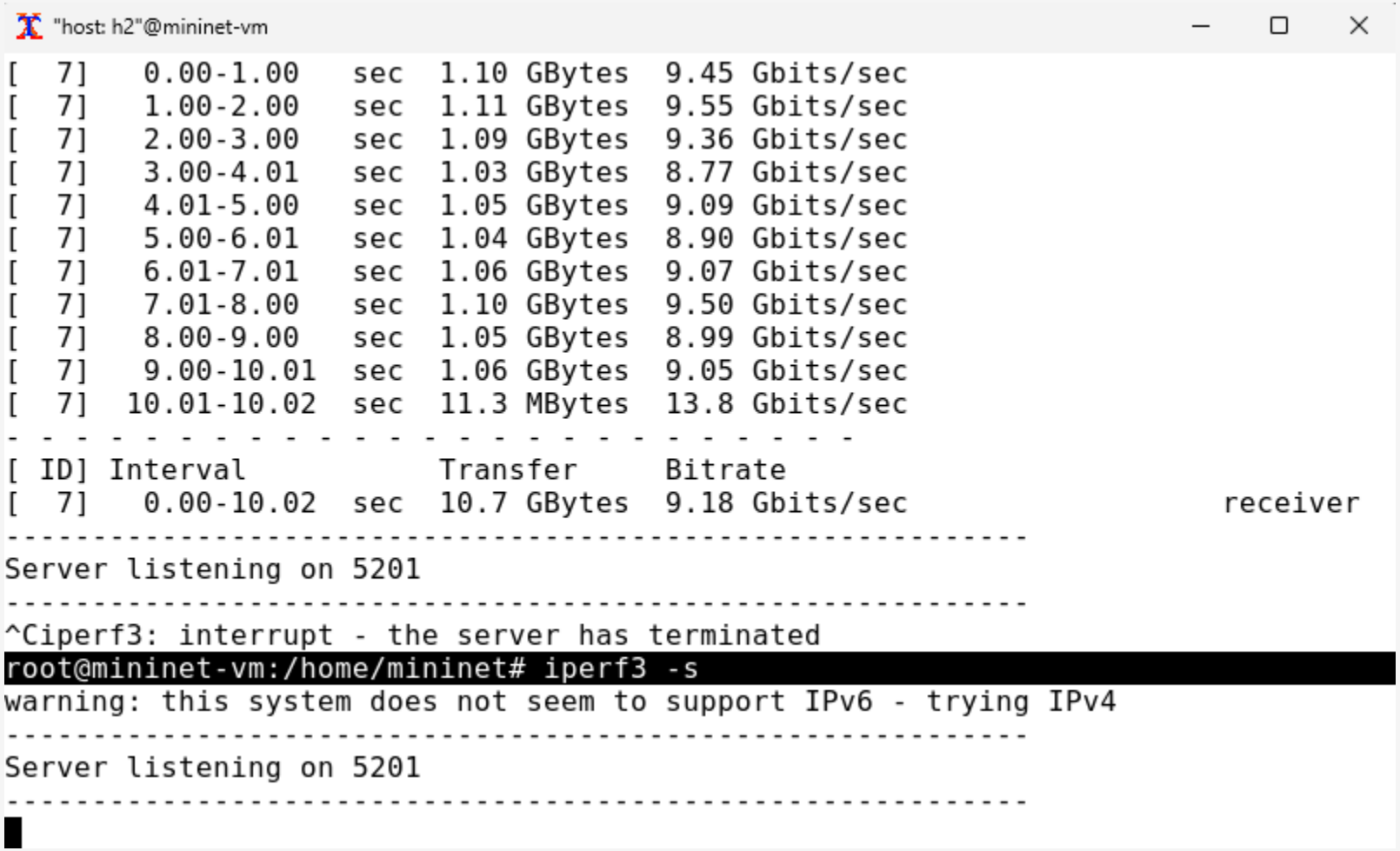

Рис. 15: Запуск iPerf3 в режиме сервера на хосте h2

Интерактивные эксперименты

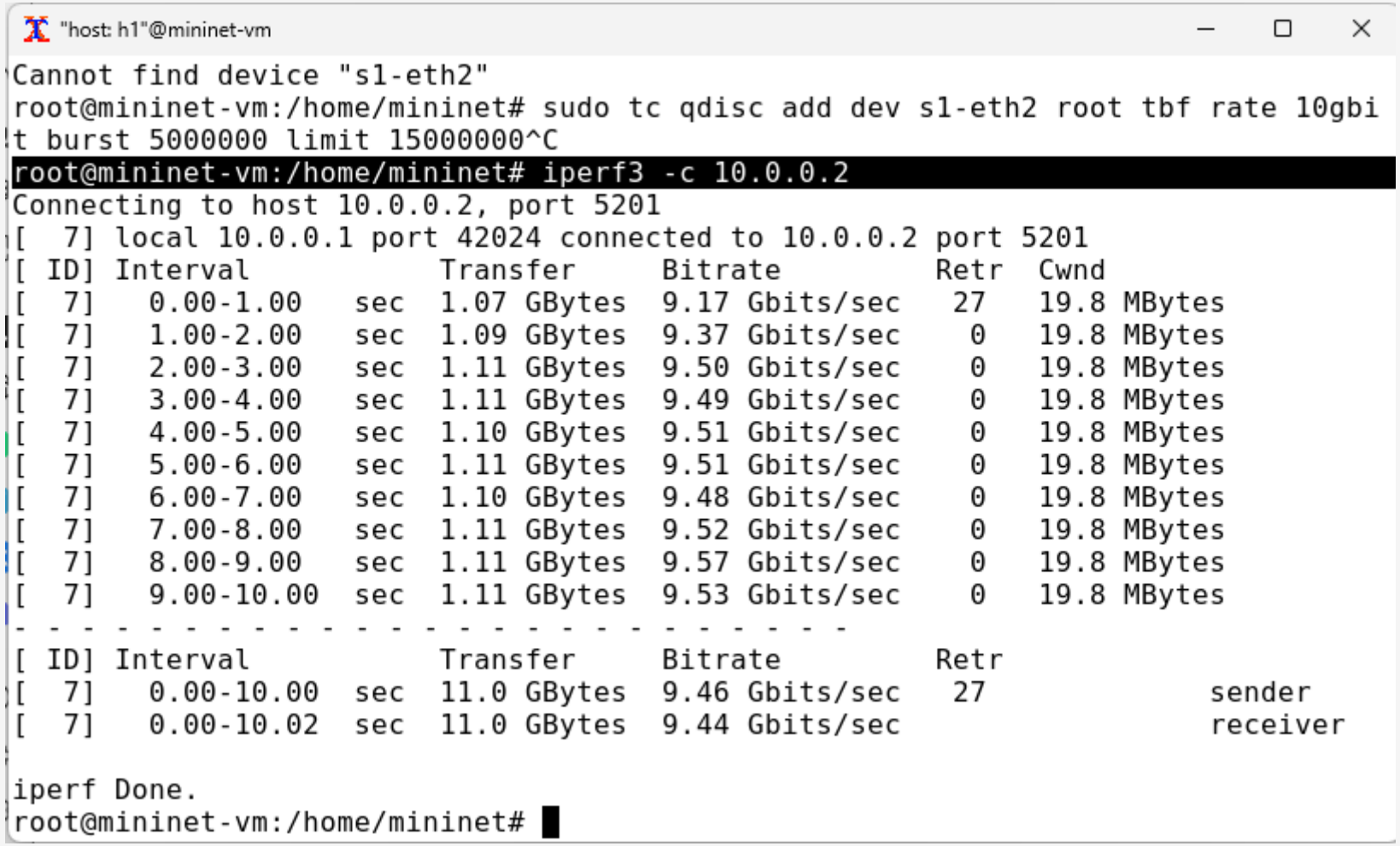

Рис. 16: Запуск iPerf3 в режиме клиента на хосте h1

Интерактивные эксперименты

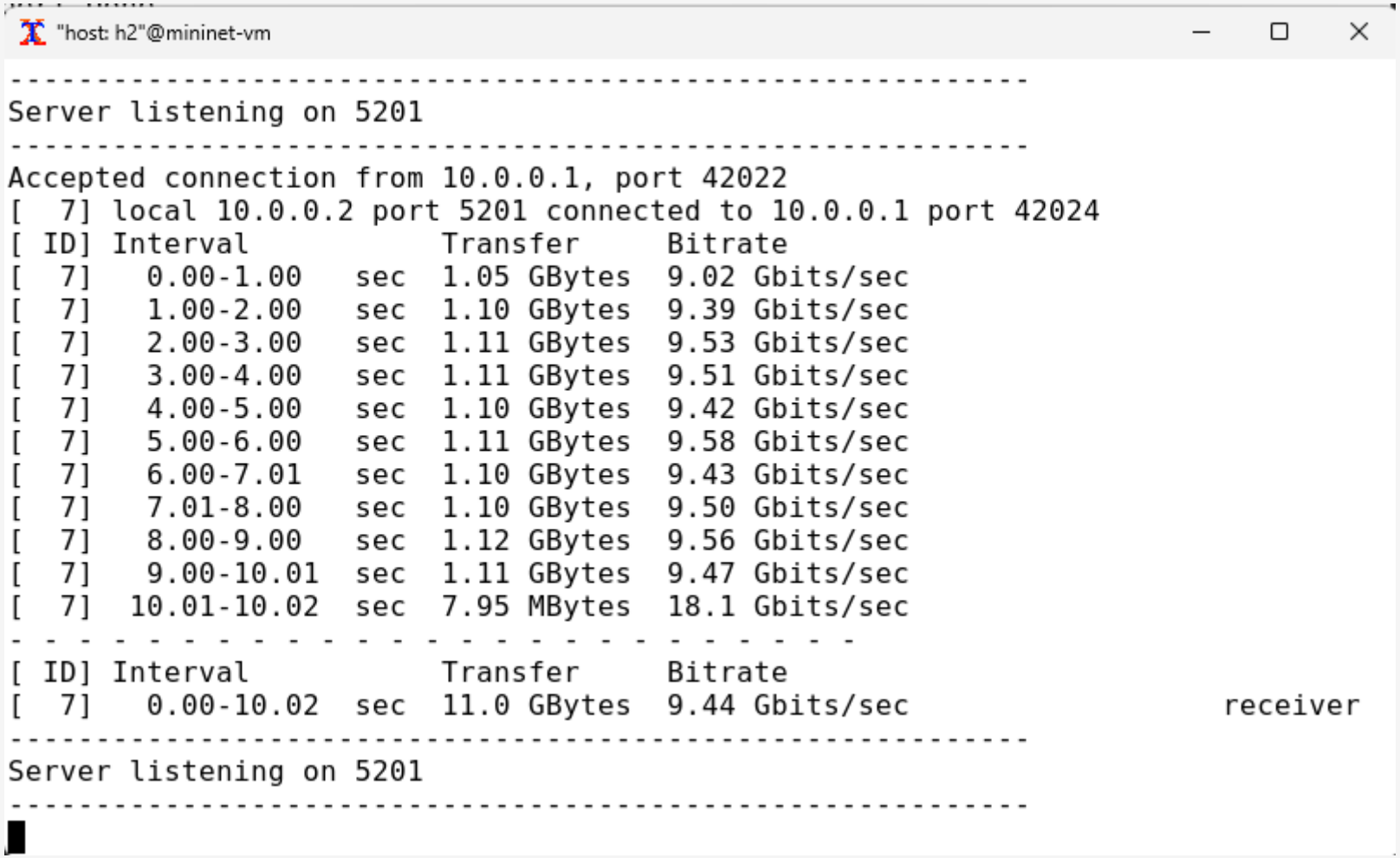

Рис. 17: Остановка iPerf3

Интерактивные эксперименты

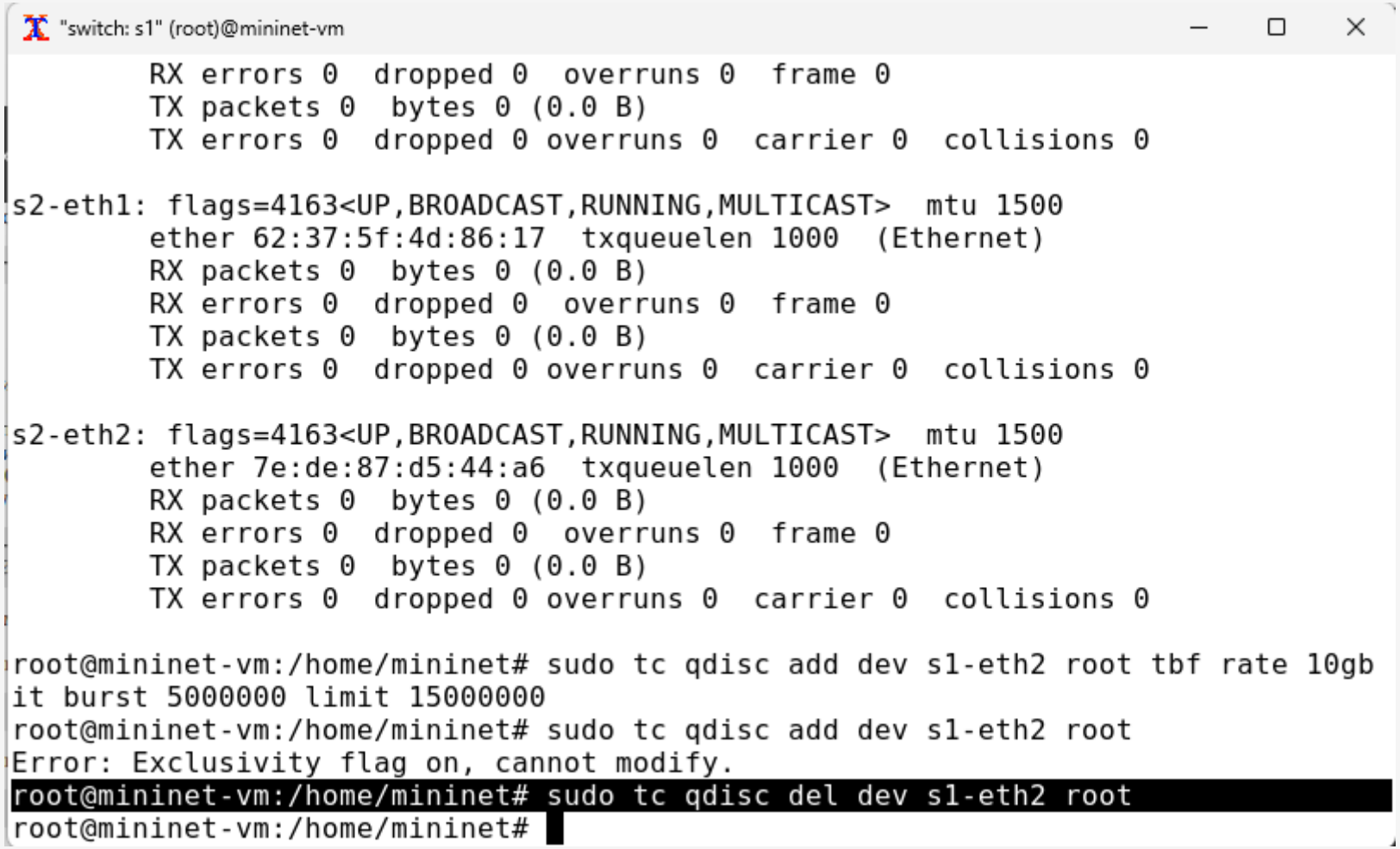

Рис. 18: Удаление модифицированной конфигурации на коммутаторе s1

Интерактивные эксперименты

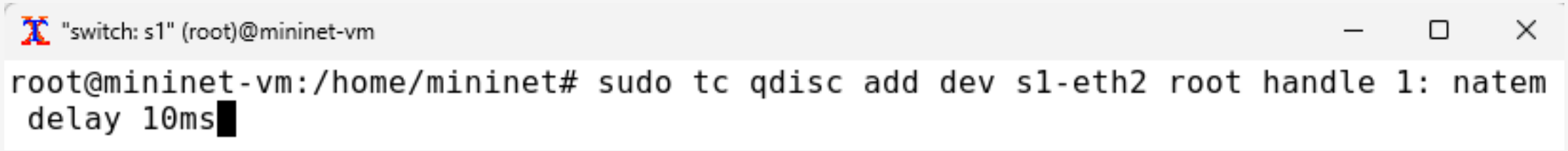

Рис. 19: Объединение NETEM и TBF

Интерактивные эксперименты

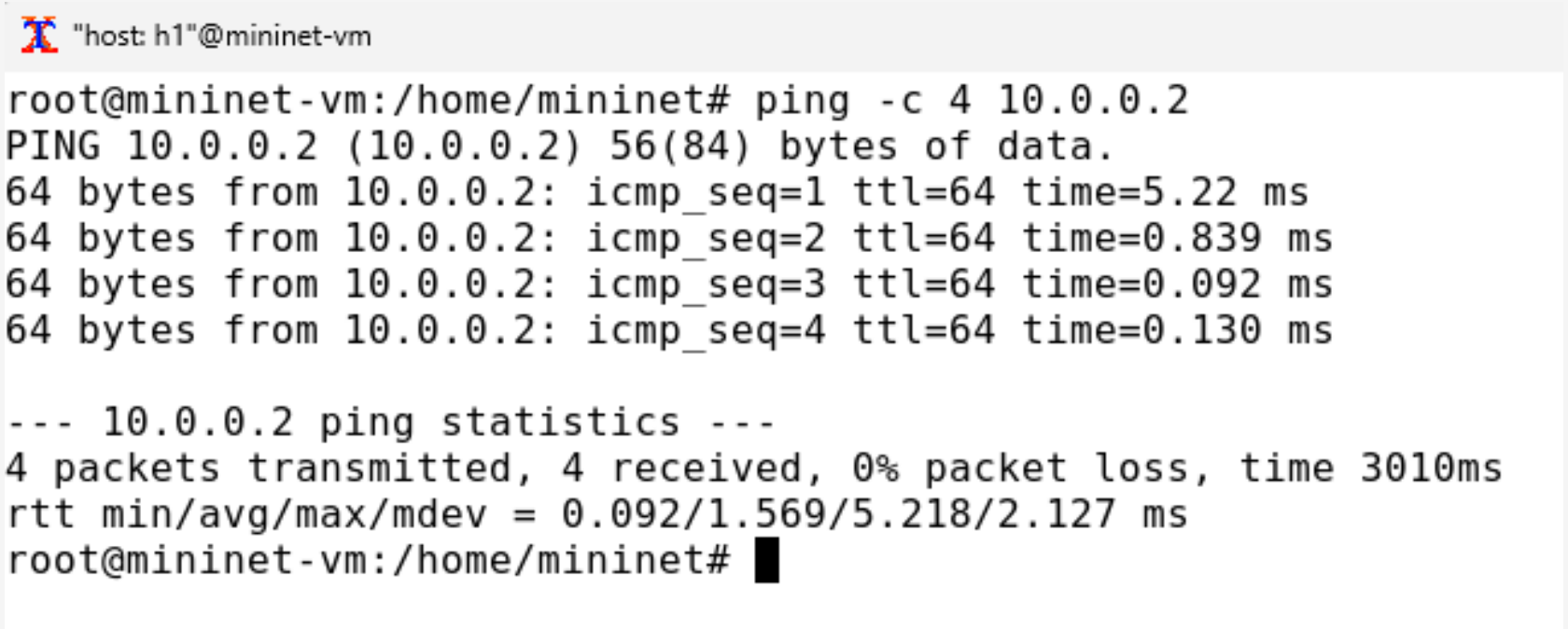

Рис. 20: Проверка задержки

Интерактивные эксперименты

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev s1-eth2 parent 1: handle 2:
tbf rate 2gbit burst 1000000 limit 2000000
root@mininet-vm:/home/mininet#
```

Рис. 21: Добавление второго правила на коммутаторе s1

Интерактивные эксперименты

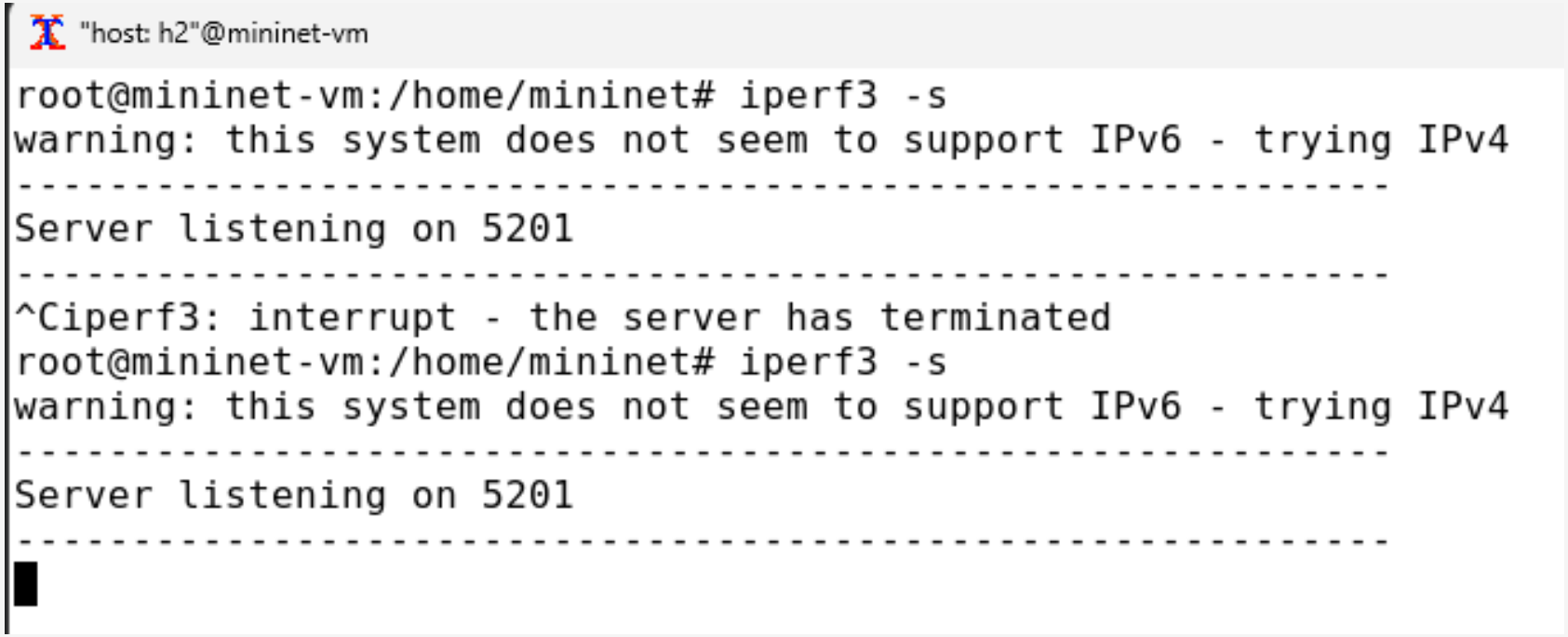

Рис. 22: Запуск iPerf3 в режиме сервера на хосте h2

Интерактивные эксперименты

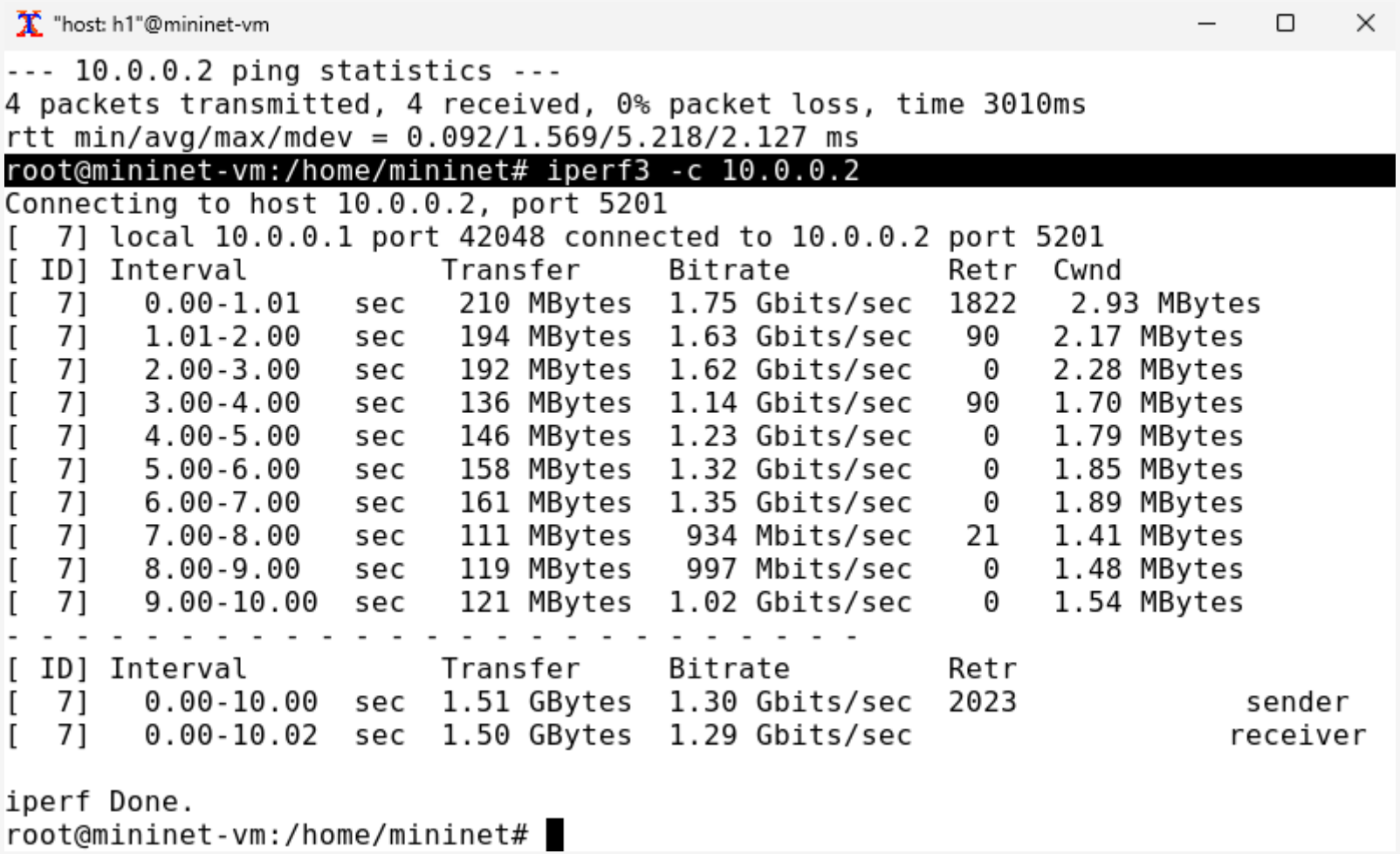

Рис. 23: Запуск iPerf3 в режиме клиента на хосте h1

Интерактивные эксперименты

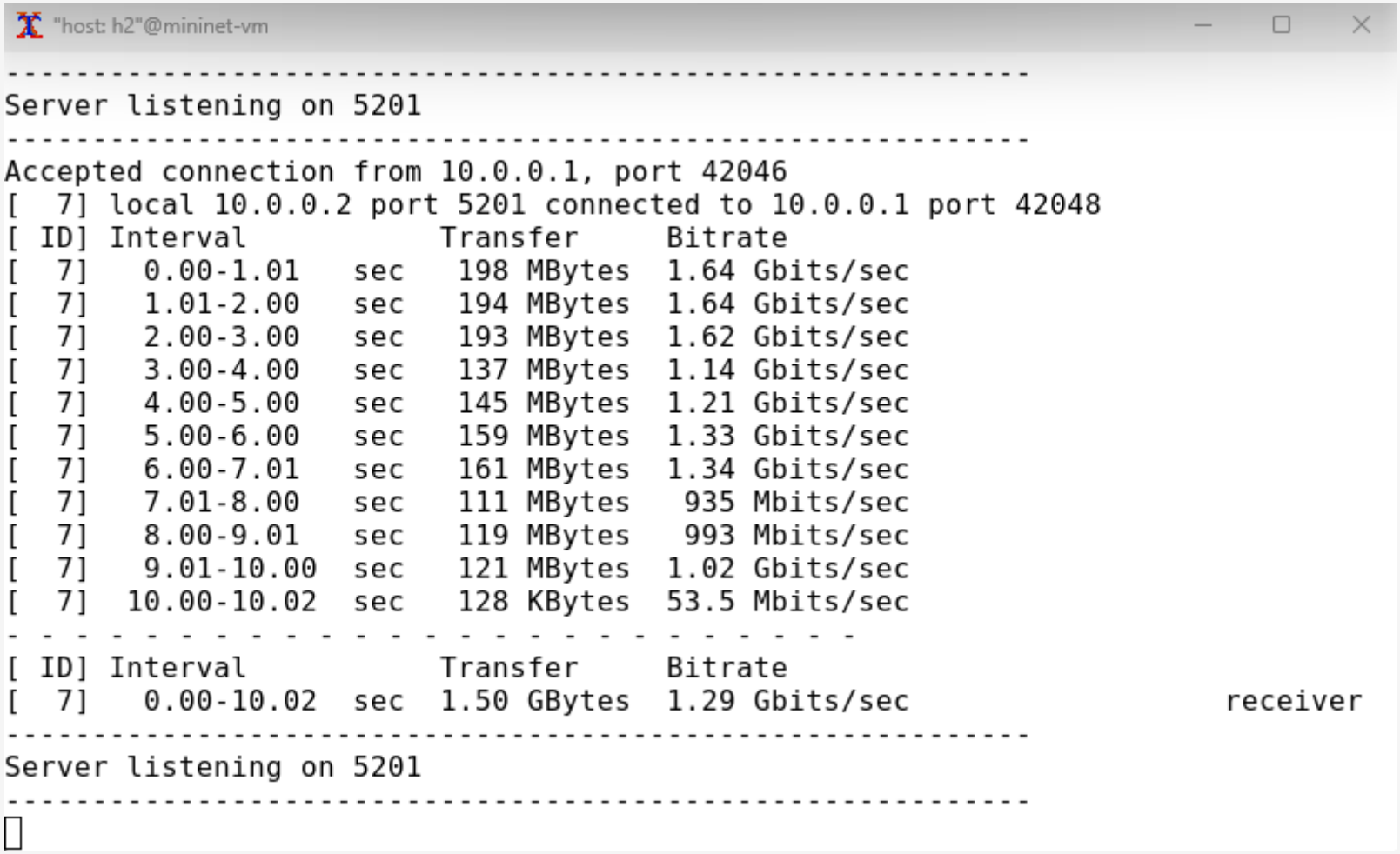

Рис. 24: Остановка iPerf3

Интерактивные эксперименты

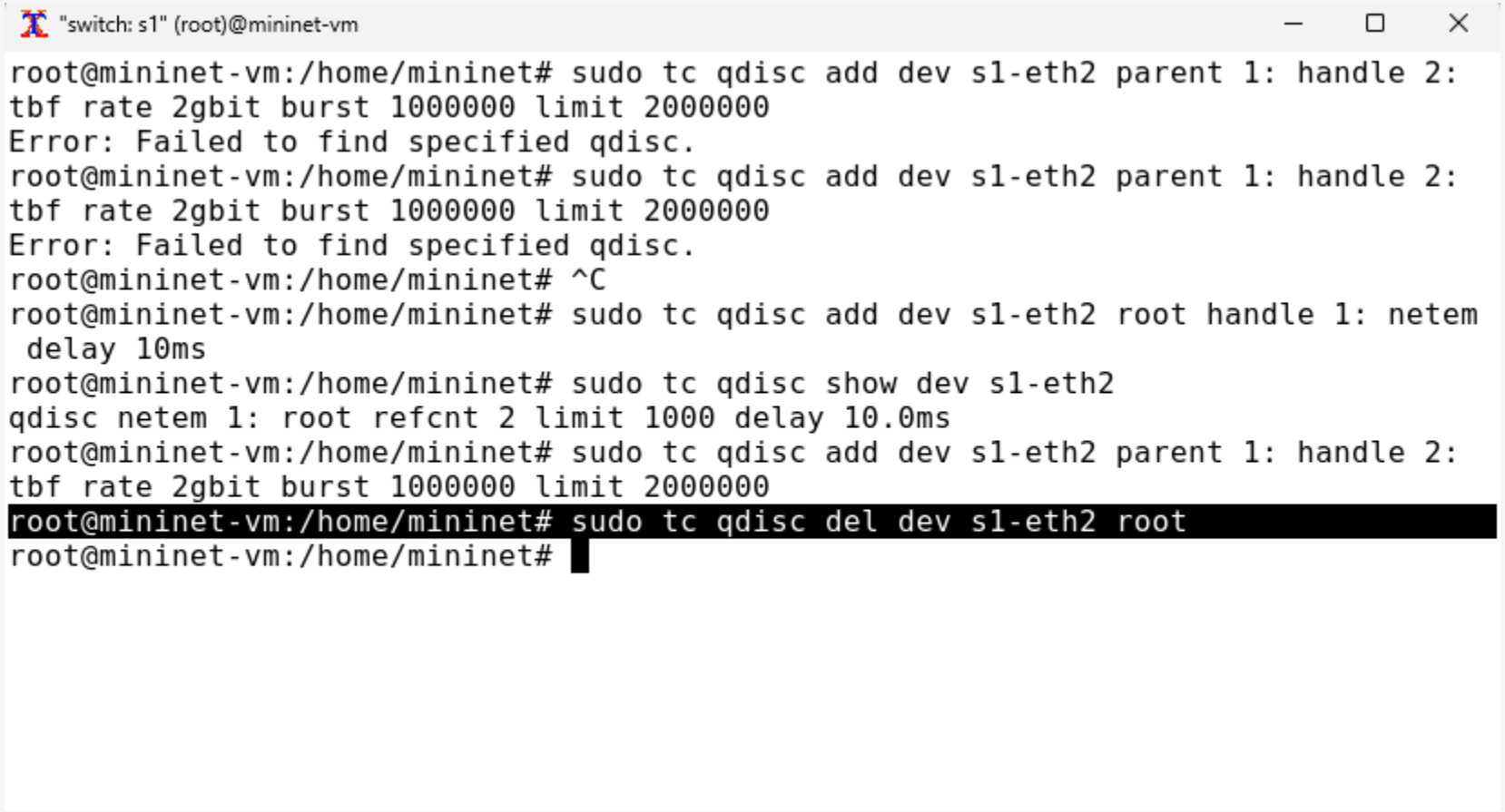

Рис. 25: Удаление модифицированной конфигурации на коммутаторе s1

Воспроизводимые эксперименты

```
mininet@mininet-vm:~$ ls
mininet  mininet.orig  oflops  oftest  openflow  pox  work
mininet@mininet-vm:~$ cd work/
mininet@mininet-vm:~/work$ ls
lab_iperf3  lab_netem_i  lab_netem_ii  lesson1.mn
mininet@mininet-vm:~/work$ mkdir lab6
mininet@mininet-vm:~/work$ ls
lab6  lab_iperf3  lab_netem_i  lab_netem_ii  lesson1.mn
mininet@mininet-vm:~/work$ cd lab6
mininet@mininet-vm:~/work/lab6$ mkdir exp1
mininet@mininet-vm:~/work/lab6$ mkdir exp2
mininet@mininet-vm:~/work/lab6$ ls
exp1  exp2
mininet@mininet-vm:~/work/lab6$
```

Рис. 26: Создание необходимых каталогов

Воспроизводимые эксперименты

mininet@mininet-vm: ~/work/lab6/exp1

```
  GNU nano 4.8                                                                ex
#!/usr/bin/env python

"""
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
"""

from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info

def emptyNet():

    "Create an empty network and add nodes to it."

    net = Mininet( controller=Controller, waitConnected=True )

    info( '*** Adding controller\n' )
    net.addController( 'c0' )

    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( '*** Adding switch\n' )
    s1 = net.addSwitch( 's1' )
    s2 = net.addSwitch( 's2' )

    s1.cmd('ip link del s1-eth2')
    s2.cmd('ip link del s2-eth1')

    info( '*** Creating links\n' )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )
    net.addLink( s1, s2 )

    info( '*** Starting network\n')
    net.start()

    s1.cmdPrint('sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms')
    s2.cmdPrint('sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000')

    info( '*** Running CLI\n' )
    h2.cmdPrint('iperf3 -s -D -1')
    time.sleep(10) # Wait 10 seconds
    h1.cmdPrint('iperf3 -s -D -1')
    h1.cmdPrint('iperf3 -s -D -1')

    info( '*** Stopping network' )
    net.stop()


if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
```

Рис. 27: Написание 1 скрпита

Воспроизводимые эксперименты

mininet@mininet-vm: ~/work/lab6/exp2

```
  GNU nano 4.8
all: ping.dat ping.png

ping.dat
        sudo python exp1.py
        suod chown mininet:mininet ping.dat

ping.png:
        ./ping_plot

clean:
        -rm -f &.dat *.png *.json
```

Рис. 28: Написание 2 скрпита

mininet@mininet-vm: ~/work/lab6/exp2

```
  GNU nano 4.8
#!/usr/bin/gnuplot --persist

set terminal png crop
set output 'ping.png'
set xlabel "Sequence number"
set ylabel "Delay (ms)"
set grid
plot "ping.dat" with lines
```

Рис. 29: Написание 3 скрпита

Воспроизводимые эксперименты

```
mininet@mininet-vm: ~/work/lab6/exp1
mininet@mininet-vm:~/work/lab6/exp1$ make
sudo python lab6_exp1.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Waiting for switches to connect
s1 s2
*** Set delay
*** s1 : ('tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms',)
*** s2 : ('tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000',)
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab6/exp1$ ls
iperf_result.json  lab6_exp1.py  Makefile  ping.dat  ping_plot  ping.png
mininet@mininet-vm:~/work/lab6/exp1$
```

Рис. 30: Выполнение скриптов

Воспроизводимые эксперименты

mininet@mininet-vm: ~/work/lab6/exp1

```
/home/mininet/work/lab6/exp1/lab6_exp1.py   [-M--] 92 L:[ 12+32  44/ 58] *(1029/1469b)   39 0x027
from mininet.log import setLogLevel, info
import time

def emptyNet():

    "Create an empty network and add nodes to it."

    net = Mininet( controller=Controller, waitConnected=True )

    info( '*** Adding controller\n' )
    net.addController( 'c0' )

    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( '*** Adding switch\n' )
    s1 = net.addSwitch( 's1' )
    s2 = net.addSwitch( 's2' )

    s1.cmd('ip link del s1-eth2')
    s2.cmd('ip link del s2-eth1')

    info( '*** Creating links\n' )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )
    net.addLink( s1, s2 )

    info( '*** Starting network\n')
    net.start()

    info( '*** Set delay\n')
    s1.cmdPrint( 'tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000' )


    info( '*** Traffic generation\n')
    h2.cmdPrint('iperf3 -s -D -1')
    time.sleep(10) # Wait 10 seconds
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json')
    h1.cmdPrint('ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')

    info( '*** Stopping network' )
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
```

Рис. 31: Изменение параметров

Воспроизводимые эксперименты

mininet@mininet-vm: ~/work/lab6/exp1

```
mininet@mininet-vm:~/work/lab6/exp1$ make
sudo python lab6_exp1.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Waiting for switches to connect
s1 s2
*** Set delay
*** s1 : ('tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000',)
Cannot find device "h1-eth0"
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab6/exp1$ ls
iperf_result.json  lab6_exp1.py  Makefile  ping.dat  ping_plot  ping.png
mininet@mininet-vm:~/work/lab6/exp1$
```

Рис. 32: Выполнение скриптов

Воспроизводимые эксперименты

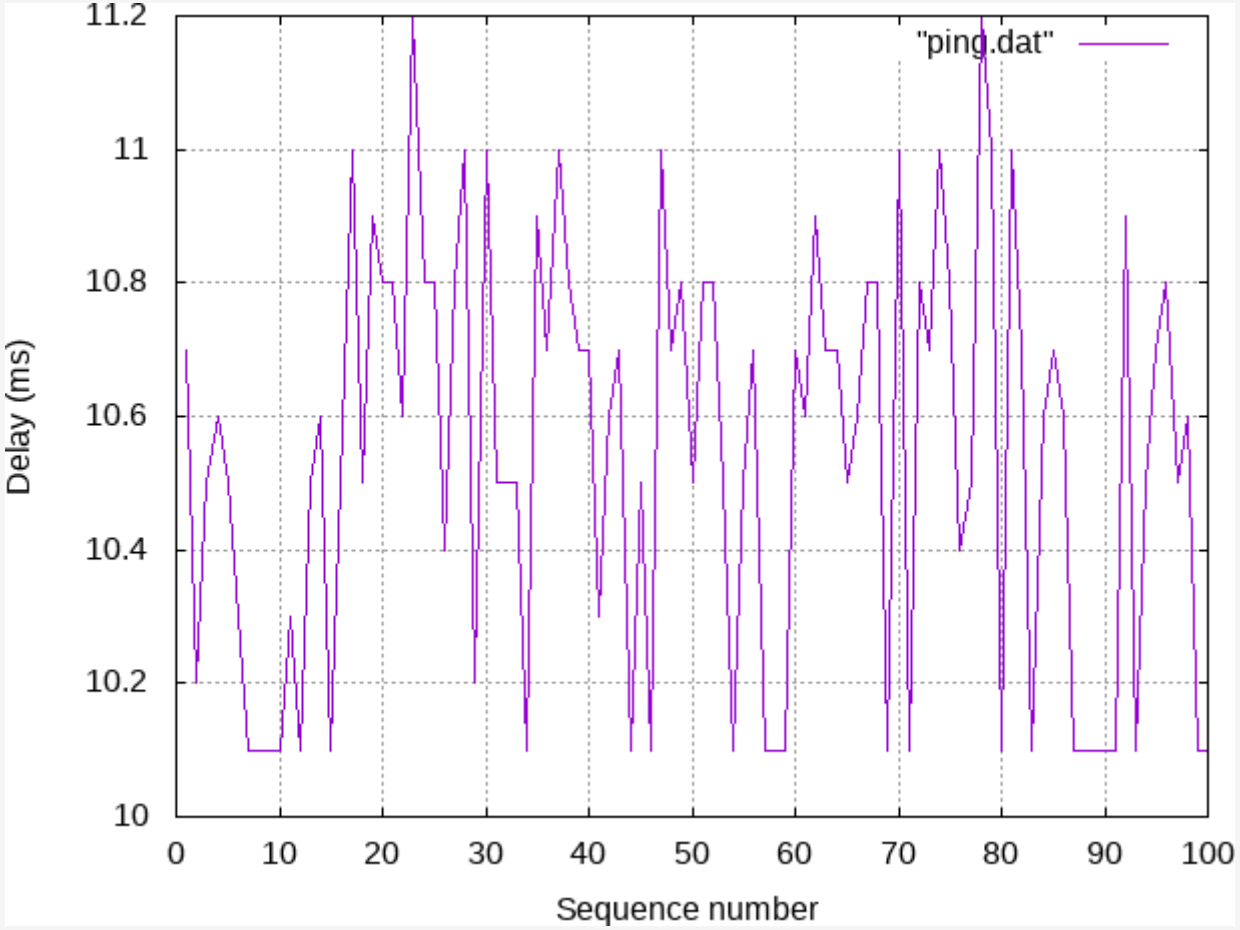

Рис. 33: График №1

Воспроизводимые эксперименты

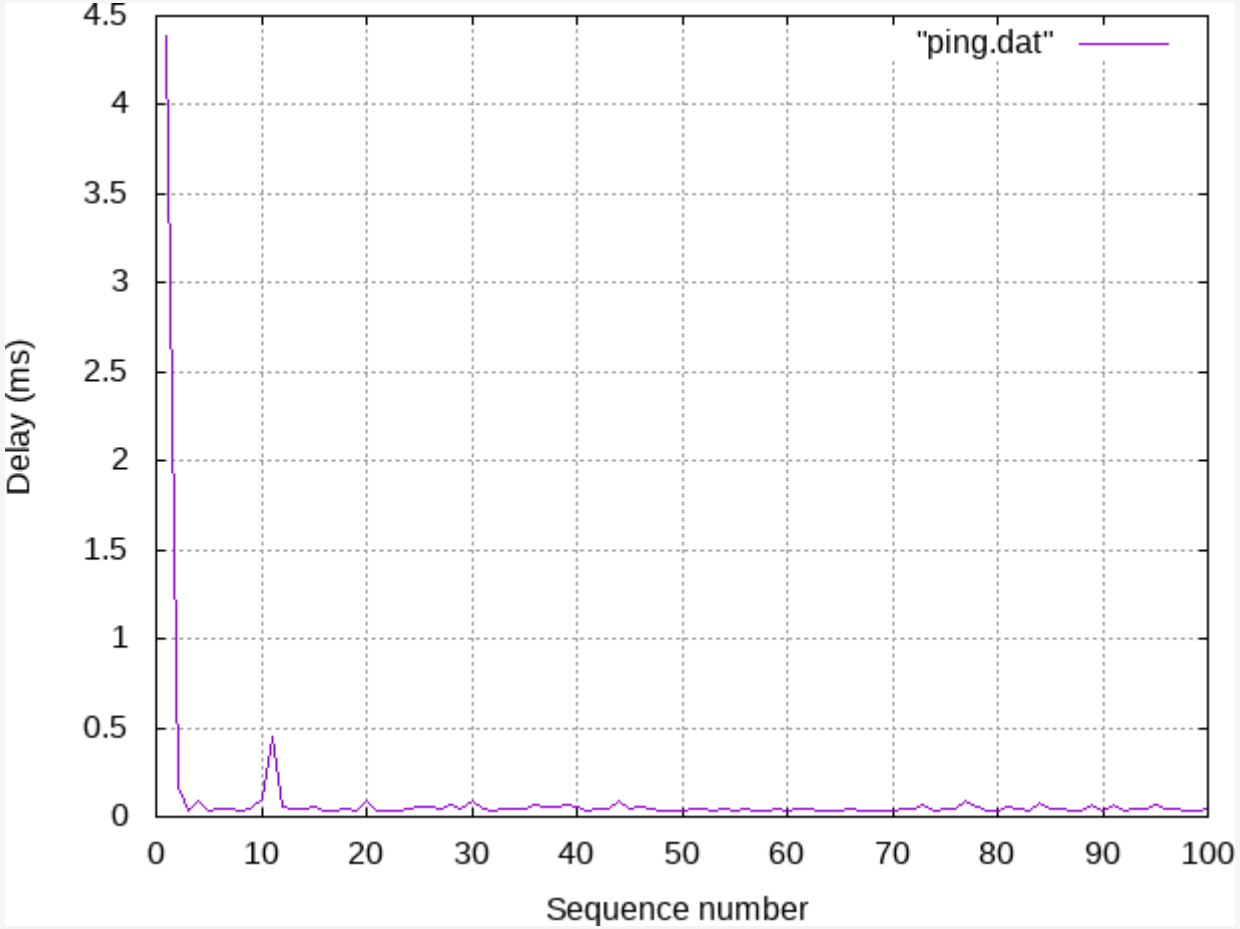

Рис. 34: График №2

Вывод

• В ходе выполнения лабораторной работы познакомились с принципами работы дисциплины очереди Token Bucket Filter, которая формирует входящий/исходящий трафик для ограничения пропускной способности, а также получили навыки моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.

Список литературы. Библиография

[1] Mininet: https://mininet.org/