Лабораторная работа №3

Измерение и тестирование пропускной способности сети. Воспроизводимый эксперимент

Ким Реачна[1]

29 ноября, 2023, Москва, Россия

[1]Российский Университет Дружбы Народов

Цели и задачи

Цель лабораторной работы

Основной целью работы является знакомство с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получение навыков проведения воспроизводимого эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.

Задание

1. Воспроизвести посредством API Mininet эксперименты по измерению пропускной способности с помощью iPerf3.
2. Построить графики по проведённому эксперименту.

Процесс выполнения лабораторной работы

Скрипт lab_iperf3_topo.py

mc [mininet@mininet-vm]:~/work/lab_iperf3/lab_iperf3_topo

GNU nano 4.8 /home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo.py Modified

```
#!/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' )
    s3 = net.addSwitch( 's3' )

    info( '*** Creating links\n' )
    net.addLink( h1, s3 )
    net.addLink( h2, s3 )

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

    print("Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC())
    print("Host", h2.name, "has IP address", h2.IP(), "and MAC address", h2.MAC())

    info( '*** Running CLI\n' )
    CLI( net )

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


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

Рис. 1: Скрипт lab_iperf3_topo.py

Запуск скрипта lab_iperf3_topo.py

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s3 ...
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 16:4e:6d:7c:17:ba
Host h2 has IP address 10.0.0.2 and MAC address 76:95:f4:a4:c8:a2
*** Running CLI
*** Starting CLI:
```

Рис. 2: Запуск скрипта lab_iperf3_topo.py

Скрипт lab_iperf3_topo2.py

mc [mininet@mininet-vm]:~/work/lab_iperf3/lab_iperf3_topo

GNU nano 4.8 /home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo2.py Modified

```
#!/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
from mininet.node import CPULimitedHost
from mininet.link import TCLink

def emptyNet():

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

    net = Mininet( controller=Controller, waitConnected=True, host = CPULimitedHost, link = TCLink )

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

    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1', cpu=50 )
    h2 = net.addHost( 'h2', ip='10.0.0.2', cpu=45 )

    info( '*** Adding switch\n' )
    s3 = net.addSwitch( 's3' )

    info( '*** Creating links\n' )
    net.addLink( h1, s3, bw=10, delay='5ms', max_queue_size=1000, loss=10, use_htb=True )
    net.addLink( h2, s3 )

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

    print("Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC())
    print("Host", h2.name, "has IP address", h2.IP(), "and MAC address", h2.MAC())

    info( '*** Running CLI\n' )
    CLI( net )

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


if __name__ == '__main__':
```

Рис. 3: Скрипт lab_iperf3_topo2.py

Запуск скрипта lab_iperf3_topo2.py

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo2.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(10.00Mbit 5ms delay 10.00000% loss) (10.00Mbit 5ms delay 10.00000% loss) *** Starting network
*** Configuring hosts
h1 (cfs 5000000/100000us) h2 (cfs 4500000/100000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (10.00Mbit 5ms delay 10.00000% loss) ...(10.00Mbit 5ms delay 10.00000% loss)
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 6a:47:20:0a:49:0e
Host h2 has IP address 10.0.0.2 and MAC address b2:29:0b:e5:c8:bb
*** Running CLI
*** Starting CLI:
mininet> h1 ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1042 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=22.2 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=11.0 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=11.8 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.5 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=11.6 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=10.9 ms
^C
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 8 received, 20% packet loss, time 9040ms
rtt min/avg/max/mdev = 10.545/141.396/1042.345/340.546 ms, pipe 2
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
(cfs -1/100000us) (cfs -1/100000us) *** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
```

Рис. 4: Запуск скрипт создания топологии lab_iperf3_topo2.py

Скрипт lab_iperf3.py

mc [mininet@mininet-vm]:~/work/lab_iperf3/iperf3

GNU nano 4.8 /home/mininet/work/lab_iperf3/iperf3/lab_iperf3.py Modified

```
#!/usr/bin/env python

"""
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
"""
import time
from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.node import CPULimitedHost
from mininet.link import TCLink

def emptyNet():

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

    net = Mininet( controller=Controller, waitConnected=True, host = CPULimitedHost, link = TCLink )

    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' )
    s3 = net.addSwitch( 's3' )

    info( '*** Creating links\n' )
    net.addLink( h1, s3, bw=100, delay='75ms')
    net.addLink( h2, s3 )

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

    info( '*** Traffic generation\n' )
    h2.cmdPrint( 'iperf3 -s -D -1' )
    time.sleep(10) # Wait 10 seconds for servers to start
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json')

    print("Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC())
    print("Host", h2.name, "has IP address", h2.IP(), "and MAC address", h2.MAC())

    info( '*** Running CLI\n' )
    CLI( net )
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo M-6 Copy Text

Рис. 5: Скрипт lab_iperf3.py

Запуск скрипта lab_iperf3.py

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ sudo python lab_iperf3.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(100.00Mbit 75ms delay) (100.00Mbit 75ms delay) *** Starting network
*** Configuring hosts
h1 (cfs -1/100000us) h2 (cfs -1/100000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay) ...(100.00Mbit 75ms delay)
*** Waiting for switches to connect
s3
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
Host h1 has IP address 10.0.0.1 and MAC address 5e:51:97:37:22:ed
Host h2 has IP address 10.0.0.2 and MAC address a6:91:8a:6c:9a:f5
*** Running CLI
*** Starting CLI:
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ mc

mininet@mininet-vm:~/work/lab_iperf3/iperf3$ plot_iperf.sh iperf_result.json
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ mc

mininet@mininet-vm:~/work/lab_iperf3/iperf3$ touch Makefile
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 6: Запуск скрипта lab_iperf3.py

Makefile

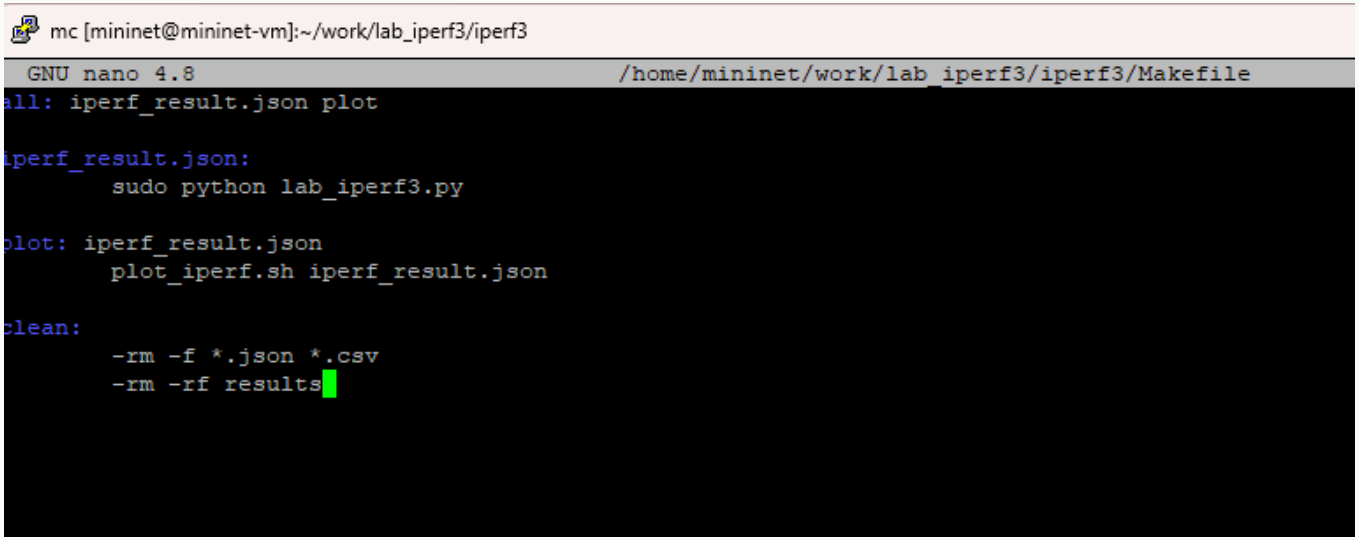

Рис. 7: Создание Makefile

Проверка корректность отработки Makefile

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ make clean
rm -f *.json *.csv
rm -rf results
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ make
sudo python lab_iperf3.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(100.00Mbit 75ms delay) (100.00Mbit 75ms delay) *** Starting network
*** Configuring hosts
h1 (cfs -1/100000us) h2 (cfs -1/100000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay) ...(100.00Mbit 75ms delay)
*** Waiting for switches to connect
s3
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
Host h1 has IP address 10.0.0.1 and MAC address ae:a7:7c:b2:40:2f
Host h2 has IP address 10.0.0.2 and MAC address d2:64:1b:78:0d:f4
*** Running CLI
*** Starting CLI:
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
plot_iperf.sh iperf_result.json
```

Рис. 8: Проверка корректность отработки Makefile

Визуализация результатов эксперимента

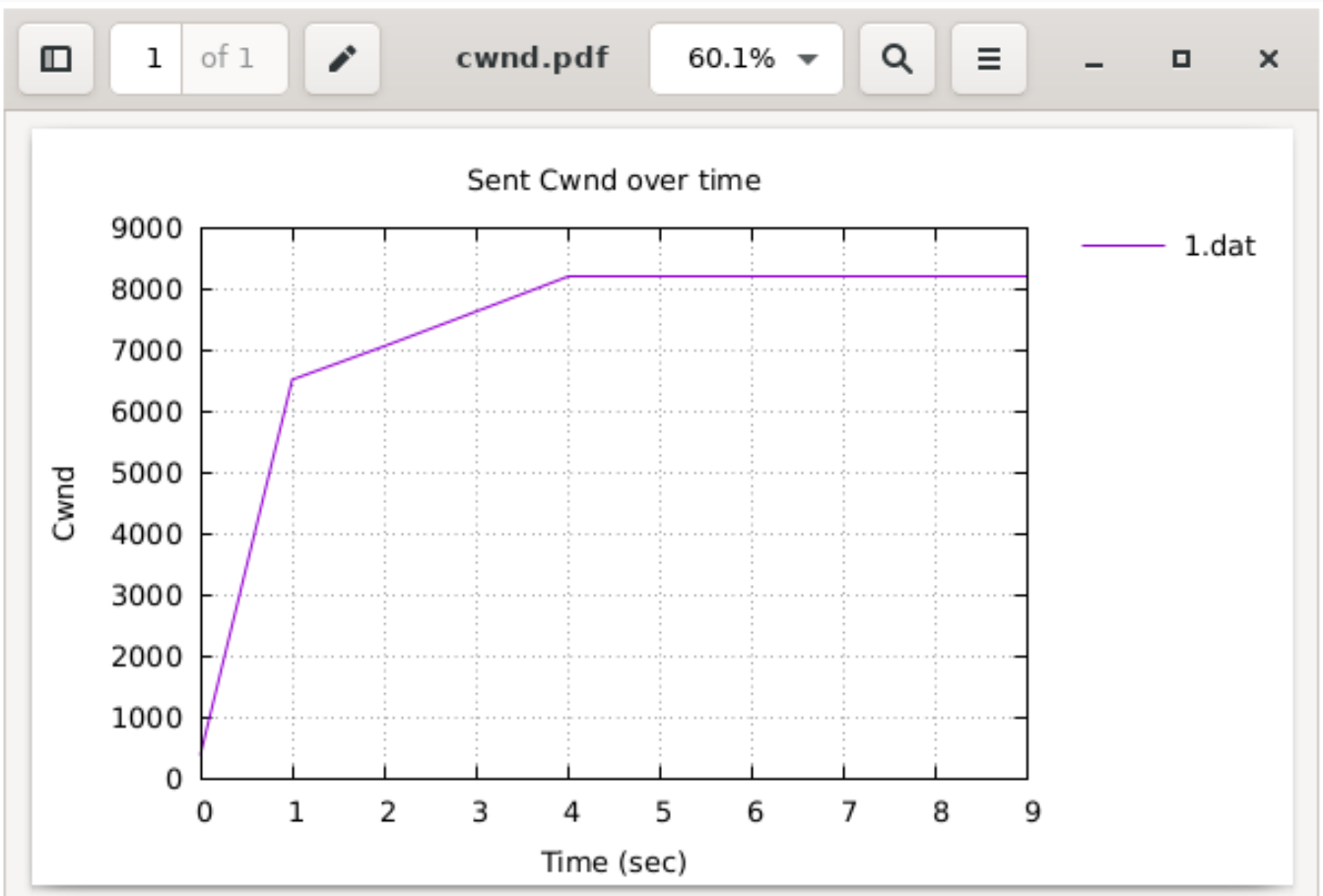

Рис. 9: Окно перегрузки

Визуализация результатов эксперимента

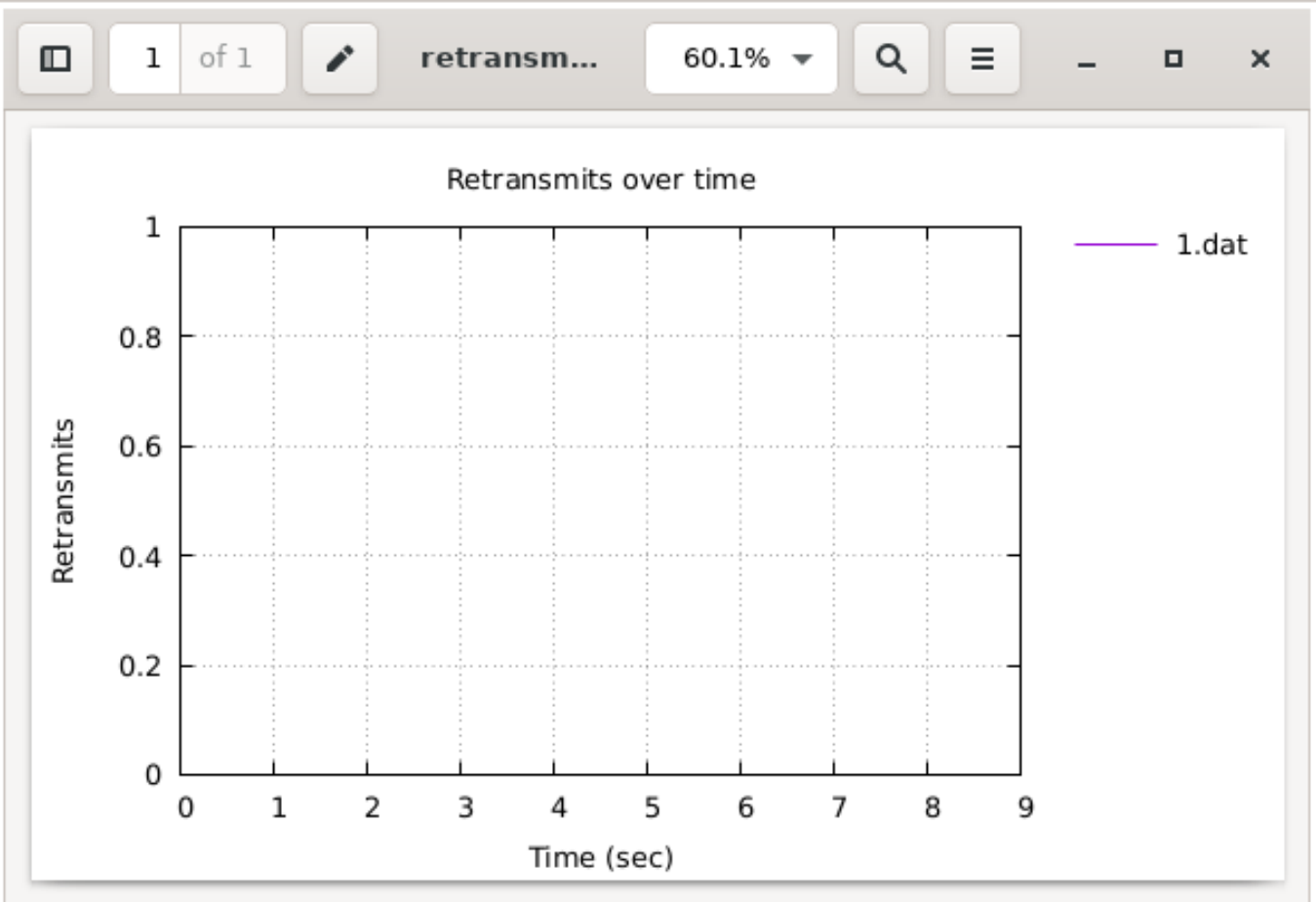

Рис. 10: Повторная передача

Визуализация результатов эксперимента

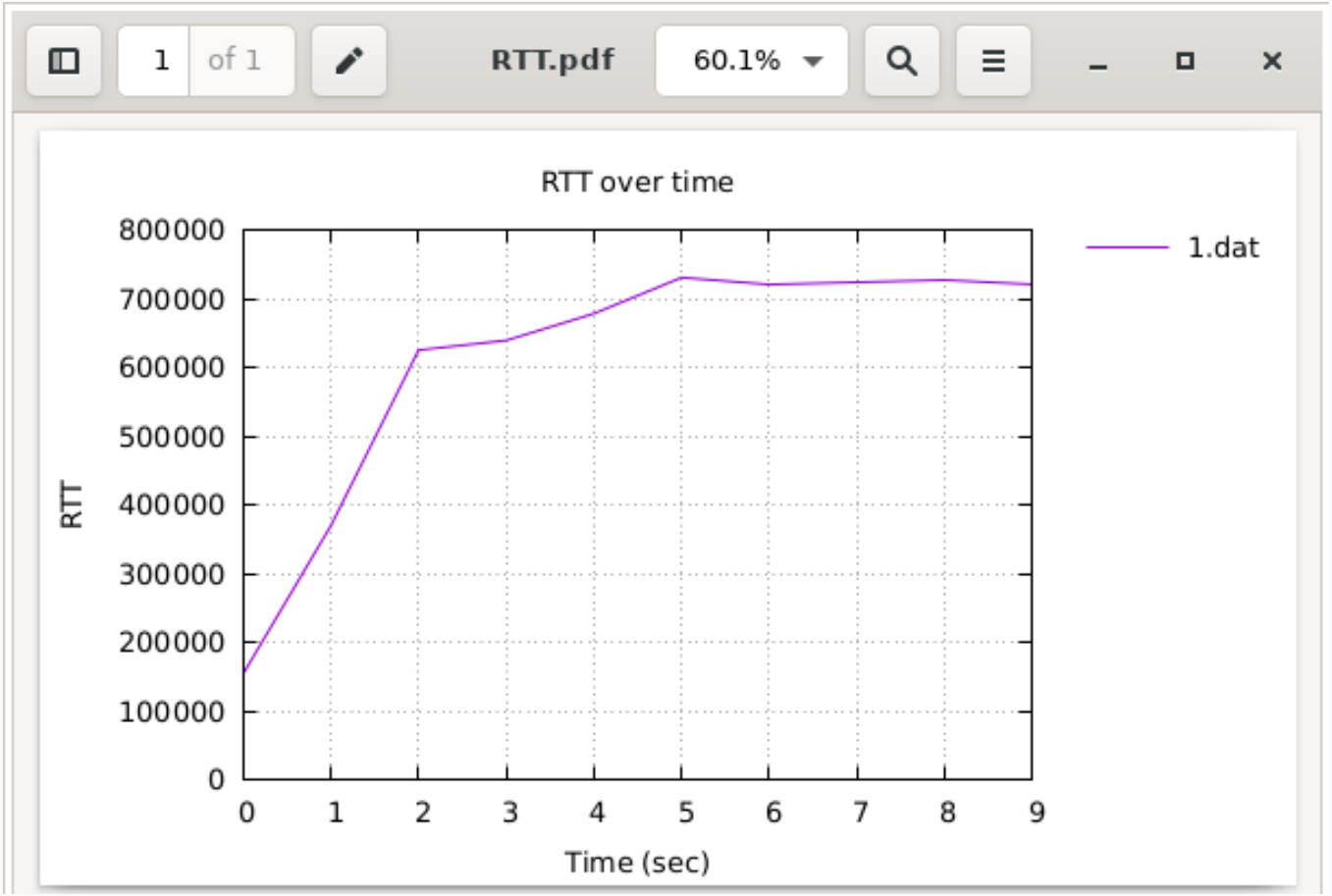

Рис. 11: Время приема-передачи

Визуализация результатов эксперимента

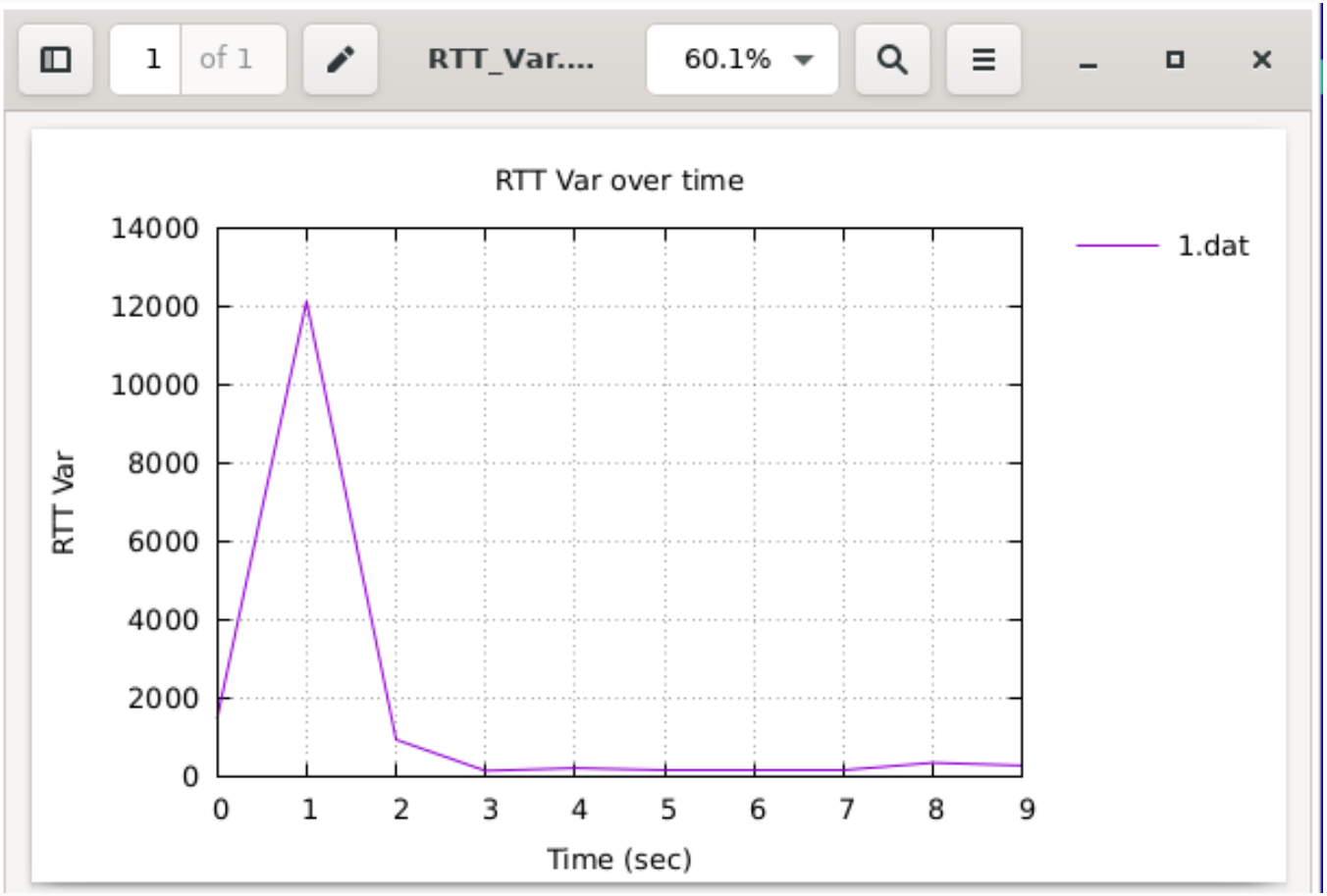

Рис. 12: Отклонение времени приема-передачи

Визуализация результатов эксперимента

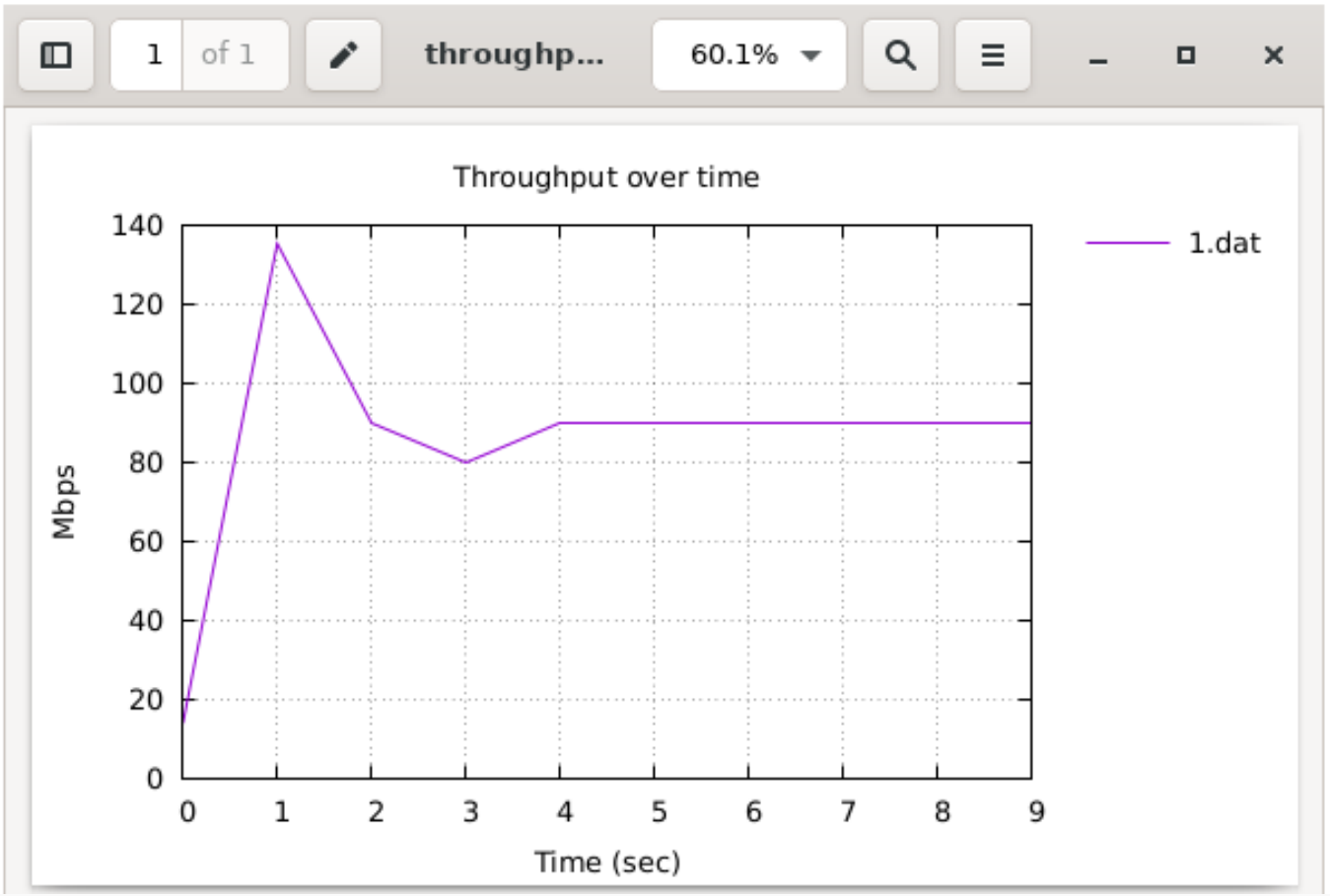

Рис. 13: Пропускная способность

Визуализация результатов эксперимента

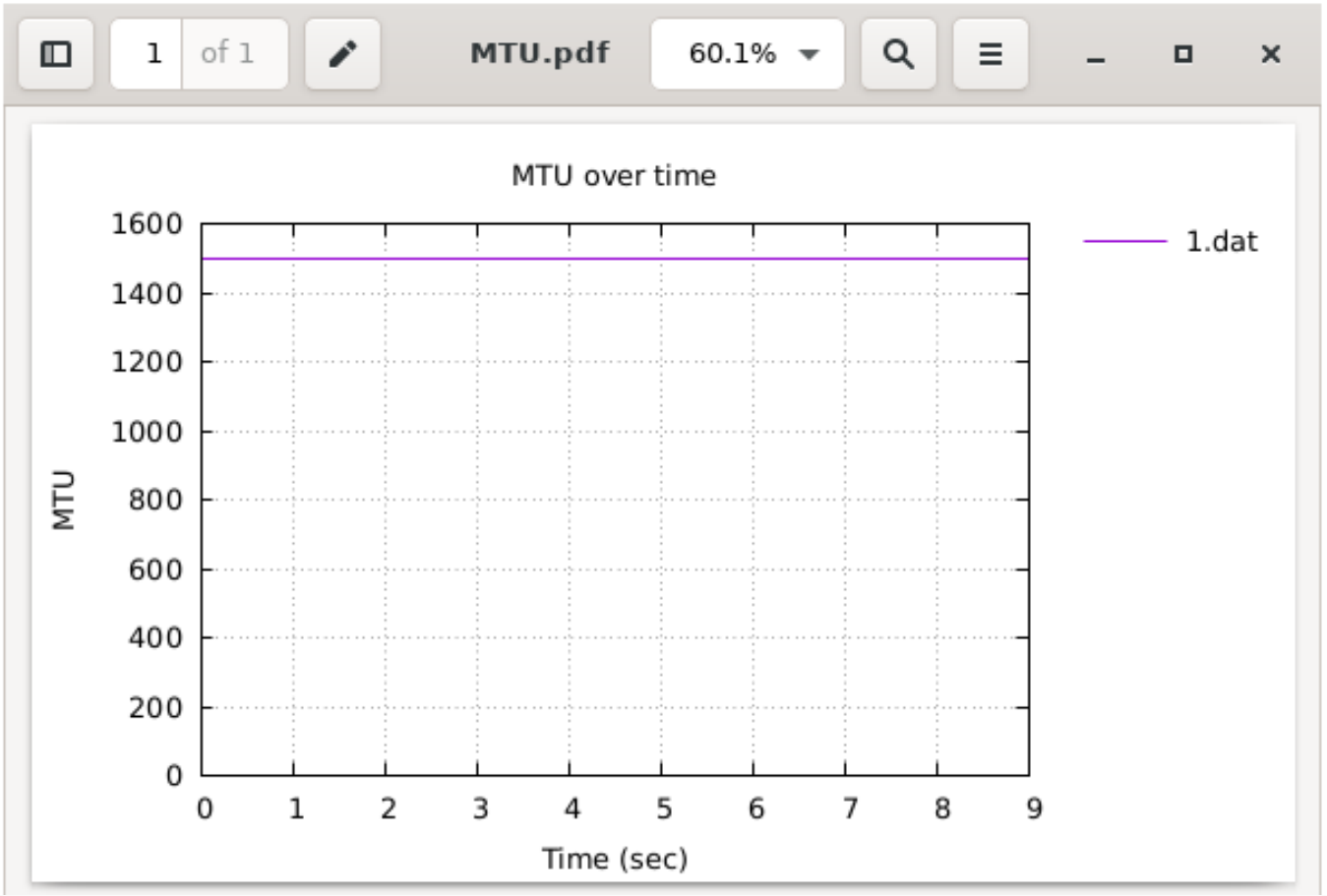

Рис. 14: Максимальная единица передачи

Визуализация результатов эксперимента

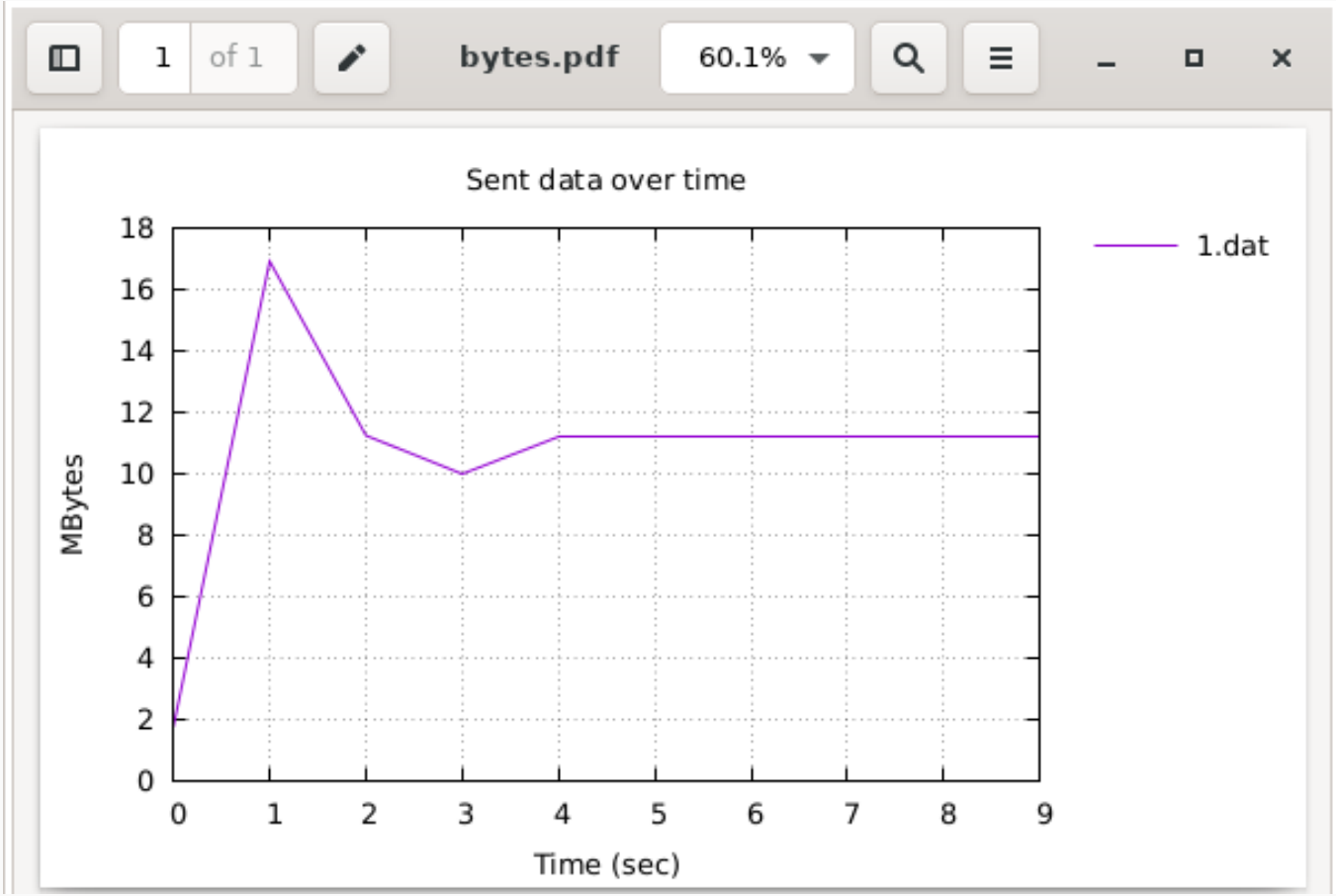

Рис. 15: Количество переданных байтов

Выводы по проделанной работе

Вывод

Я познакомилась с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получение навыков проведения воспроизводимого эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.