

Отчёт по лабораторной работе №3

**Измерение и тестирование пропускной способности сети.
Воспроизводимый эксперимент**

Ким Реачна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Листинги программы	17
4	Вывод	23

Список иллюстраций

2.1	Создание подкаталога и копирование скрипта lab_iperf3_topo.py .	5
2.2	Скрипт lab_iperf3_topo.py	6
2.3	Запуск скрипта lab_iperf3_topo.py до изменением	7
2.4	Проверка корректность отработки после изменением	7
2.5	Скрипт lab_iperf3_topo2.py	8
2.6	Запуск скрипт создания топологии lab_iperf3_topo2.py	9
2.7	Запуск скрипт создания топологии lab_iperf3_topo.py	10
2.8	Копирование в lab_iperf3.py	10
2.9	Скрипт lab_iperf3.py	11
2.10	Запуск скрипта lab_iperf3.py	12
2.11	Создание Makefile	12
2.12	Проверка корректность отработки Makefile	13
2.13	Окно перегрузки	13
2.14	Повторная передача	14
2.15	Время приема-передачи	14
2.16	Отклонение времени приема-передачи	15
2.17	Пропускная способность	15
2.18	Максимальная единица передачи	16
2.19	Количество переданных байтов	16

1 Цель 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.

2 Выполнение лабораторной работы

1. Создание подкаталога и копирование скрипта lab_iperf3_topo.py для запуска моделирования, затем изменение скрипта для вывода отображаемой информации о хосте, IP-адресе и MAC-адресе:

```
mininet@mininet-vm:~$ cd ~/work/lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ mkdir lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3$ cd ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp ~/mininet/examples/emphynet.py ~/work/lab_iperf3/
lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv emphynet.py lab_iperf3_topo.py
```

Рис. 2.1: Создание подкаталога и копирование скрипта 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. 2.2: Скрипт 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
*** Running CLI
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s3-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:h2-eth0
c0
mininet> links
h1-eth0<->s3-eth1 (OK OK)
h2-eth0<->s3-eth2 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=773>
<Host h2: h2-eth0:10.0.0.2 pid=776>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=781>
<Controller c0: 127.0.0.1:6653 pid=766>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done

```

Рис. 2.3: Запуск скрипта lab_iperf3_topo.py до изменением

```

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. 2.4: Проверка корректность отработки после изменением

2. Изменение топологии, скопировав в lab_iperf3_topo2.py, указание на использование ограничения производительности и изоляции, изменение функцию задания параметров виртуального хоста h1 и h2 и изменение функцию параметров соединения между хостом h1 и коммутатором s3:

```
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\n' )
    net.stop()

if __name__ == '__main__':
```

Рис. 2.5: Скрипт lab_iperf3_topo2.py

- Запуск на отработку сначала скрипт lab_iperf3_topo2.py, затем lab_iperf3_topo.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 5000000/1000000us) h2 (cfs 4500000/1000000us)
*** 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 -1/1000000us) (cfs -1/1000000us) *** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done

```

Рис. 2.6: Запуск скрипт создания топологии lab_iperf3_topo2.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 22:86:ac:72:10:8f
Host h2 has IP address 10.0.0.2 and MAC address 32:30:b2:43:7b:b8
*** 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=2.02 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.167 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.043 ms
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7142ms
rtt min/avg/max/mdev = 0.036/0.306/2.017/0.647 ms
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done

```

Рис. 2.7: Запуск скрипт создания топологии lab_iperf3_topo.py

3. Построение графика: изменение топологии, копирование в lab_iperf3.py , изменение параметров, создание Makefile для запуска всего эксперимента:

```

mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp lab_iperf3_topo2.py lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mkdir -p ~/work/lab_iperf3/iperf3
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv ~/work/lab_iperf3/lab_iperf3_topo/lab_iperf3.py ~/w
erf3
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cd ~/work/lab_iperf3/iperf3
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls -l
total 4
-rwxrwxr-x 1 mininet mininet 1342 Nov 30 02:12 lab_iperf3.py

```

Рис. 2.8: Копирование в 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 -l' )
    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 )
```

Рис. 2.9: Скрипт 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 -1/1000000us) h2 (cfs -1/1000000us)
*** 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 -l',)
*** 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. 2.10: Запуск скрипта lab_iperf3.py

```

mc [mininet@mininet-vm]:~/work/lab_iperf3/iperf3
GNU nano 4.8 /home/mininet/work/lab_iperf3/iperf3/Makefile
all: iperf_result.json plot

iperf_result.json:
    sudo python lab_iperf3.py

plot: iperf_result.json
    plot_iperf.sh iperf_result.json

clean:
    -rm -f *.json *.csv
    -rm -rf results

```

Рис. 2.11: 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 -l/1000000us) h2 (cfs -l/1000000us)
*** 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 -l',)
*** 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. 2.12: Проверка корректность обработки Makefile

4. Результат построения графика

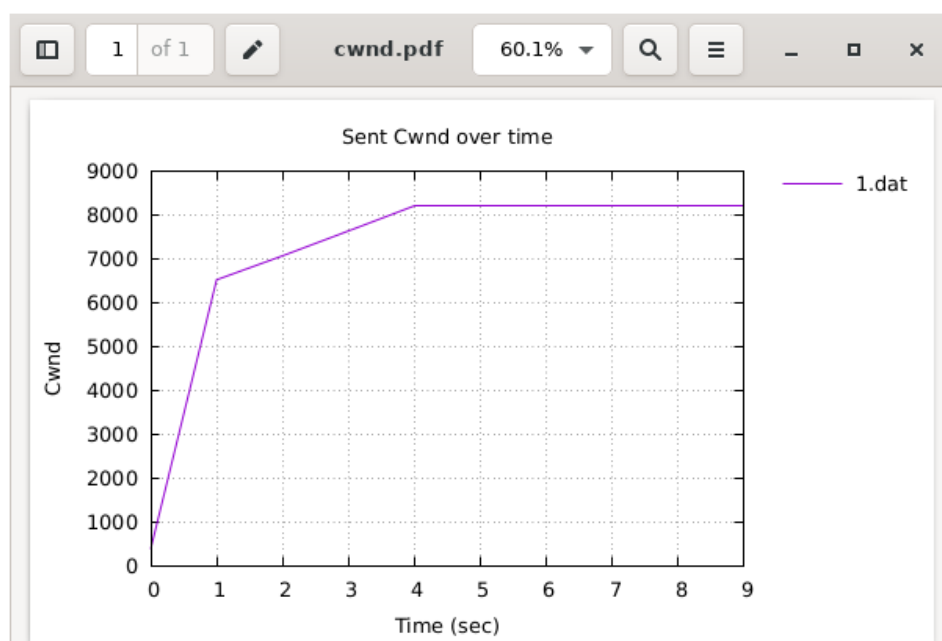


Рис. 2.13: Окно перегрузки

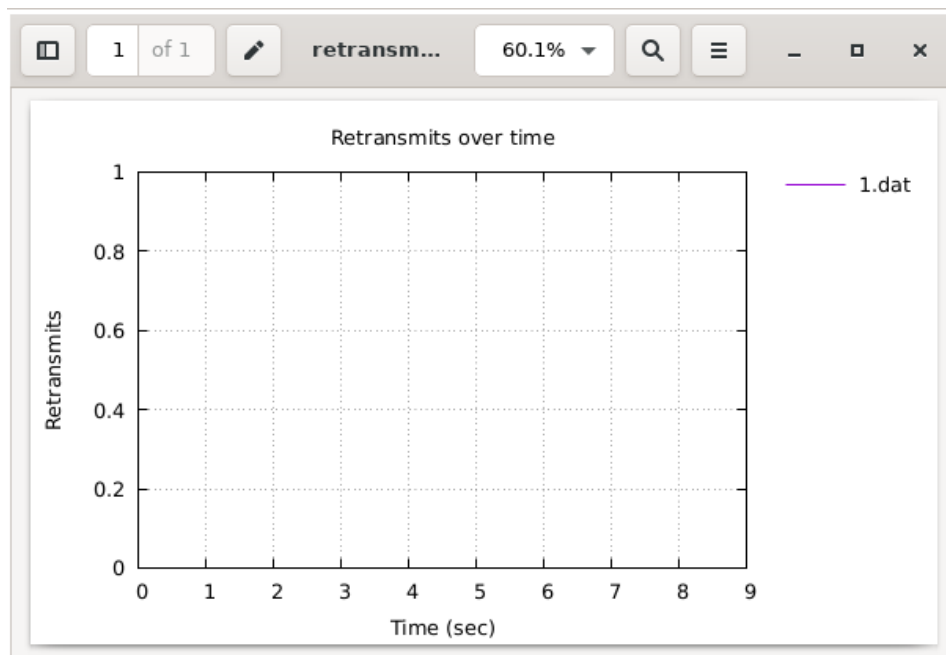


Рис. 2.14: Повторная передача

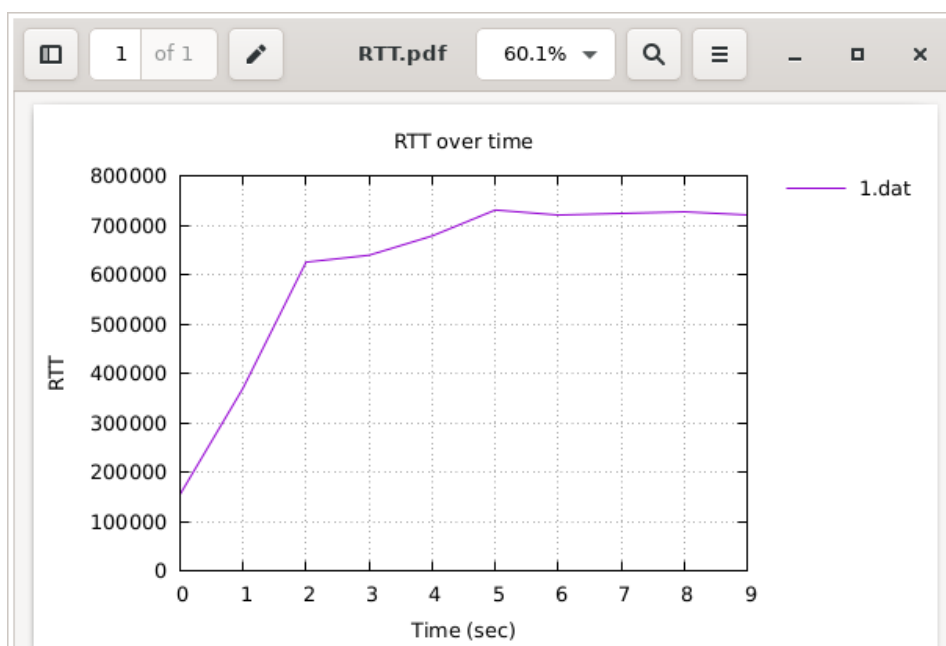


Рис. 2.15: Время приема-передачи

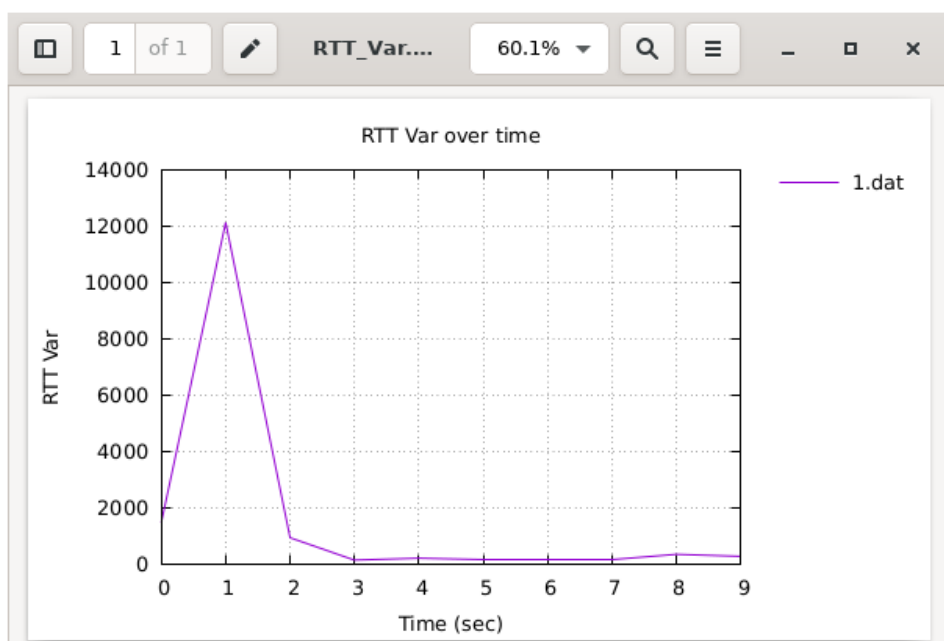


Рис. 2.16: Отклонение времени приема-передачи

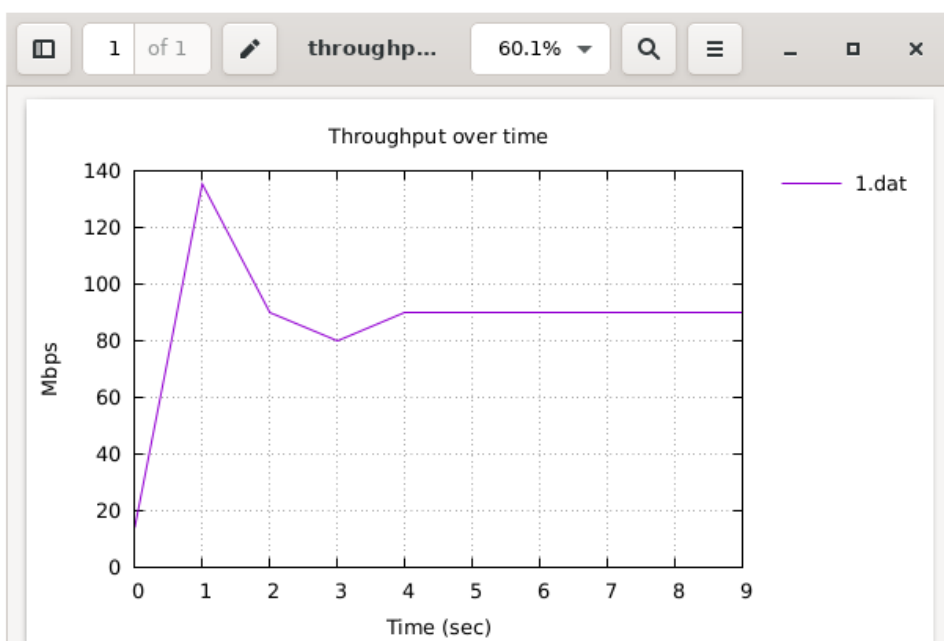


Рис. 2.17: Пропускная способность

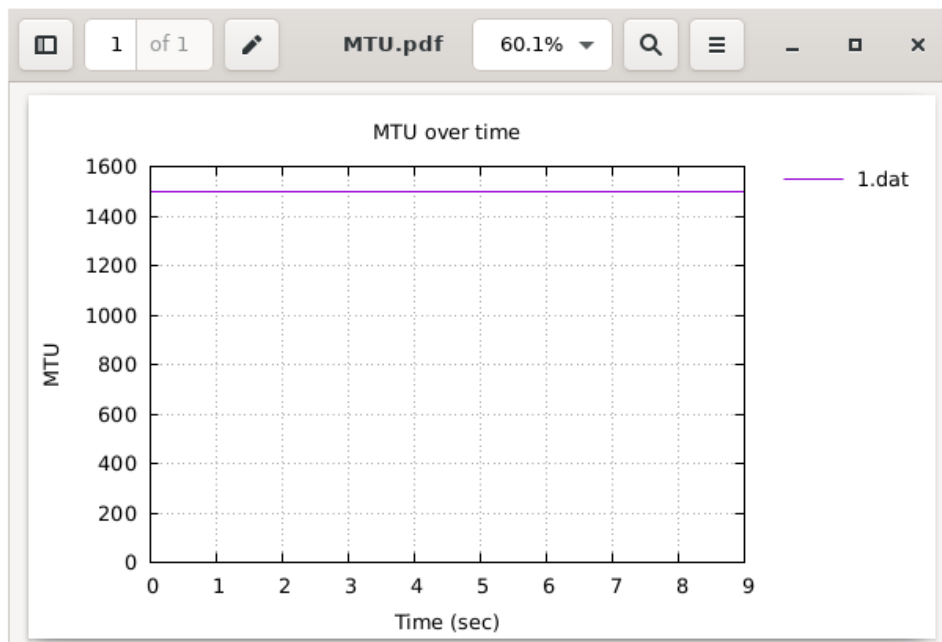


Рис. 2.18: Максимальная единица передачи

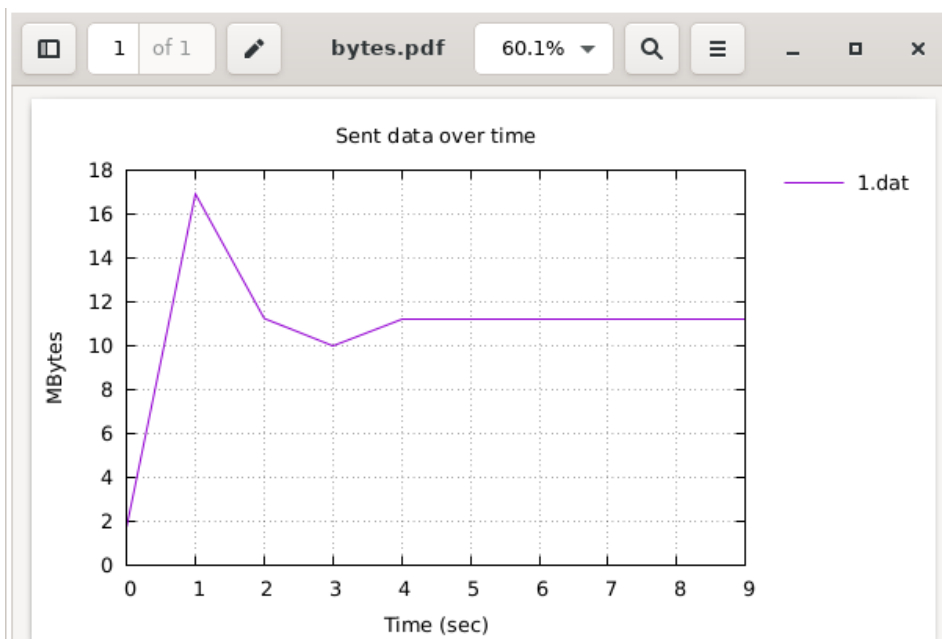


Рис. 2.19: Количество переданных байтов

3 Листинги программы

- Скрипт lab_iperf3_topo.py

```
#!/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()

```

- Скрипт lab_iperf3_topo2.py

```
#!/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__':
    setLogLevel( 'info' )
    emptyNet()

```

- Скрипт lab_iperf3.py

```
#!/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 )

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

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

```

4 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.