

# **Отчёт по научной работе**

**Ознакомление сос средством моделирования mininet**

Саргсян Арам Грачьевич

# Содержание

<b>1</b>	<b>Теоритическое введение</b>	<b>5</b>
1.1	Mininet . . . . .	5
1.2	Iperf3 . . . . .	5
<b>2</b>	<b>Процесс установки</b>	<b>7</b>
<b>3</b>	<b>Процесс моделирования</b>	<b>10</b>

# Список иллюстраций

2.1	установка mininet . . . . .	7
2.2	установка putty и xserver . . . . .	7
2.3	подключение по ssh . . . . .	8
2.4	виртуальная машина mininet . . . . .	8
2.5	Настройка соединения X11 для суперпользователя . . . . .	9
3.1	Первая модель . . . . .	10
3.2	ping от первого хоста ко второму . . . . .	11
3.3	Установка iperf3 . . . . .	11
3.4	Запуск модели . . . . .	11
3.5	Первый тест . . . . .	12
3.6	Второй тест . . . . .	12
3.7	Окно перегрузки . . . . .	13
3.8	Количество переданных байтов . . . . .	13
3.9	Топология в API Mininet . . . . .	14
3.10	Результаты . . . . .	14

## Список таблиц

# 1 Теоритическое введение

## 1.1 Mininet

Mininet — это симулятор сетевых топологий на основе виртуаилизации, который позволяет моделировать и изучать поведение сетей в контролируемой среде, основанный на использовании виртуальных машин и пространств имен Linux для создания изолированных сетевых узлов. Моделирование сетевых топологий с помощью Mininet позволяет исследовать различные сетевые протоколы, маршрутизацию, управление трафиком и т.д. Возможности моделирования с помощью Mininet включают создание виртуальных сетевых узлов, конфигурирование топологий (связь между узлами, настраивать IP-адреса, маршрутизацию), имитировать различные условия сети, такие как задержки, потери пакетов и пропускную способность, интеграция с контроллерами для исследования новых протоколов и алгоритмов.

## 1.2 Iperf3

iPerf3 представляет собой кроссплатформенное клиент-серверное приложение с открытым исходным кодом, которое можно использовать для измерения пропускной способности между двумя конечными устройствами. iPerf3 может работать с транспортными протоколами TCP, UDP и SCTP:

TCP и SCTP: - измерение пропускной способности - возможность задать размер MSS/MTU - отслеживание размера окна перегрузки TCP (CWnd)

UDP: - измерение пропускной способности - измерение потери пакетов - измерение колебания задержки (jitter) - поддержка групповой рассылки пакетов (multicast).

## 2 Процесс установки

1. Для работы с данным средством я скачал рекомендуемый к установке образ виртуальной машины: mininet-2.3.0- 210211-ubuntu-20.04.1-legacy-server-amd64-ovf. Для работы из под операционной системы Windows дополнительно установил Putty и VcXsrv Windows X Server.

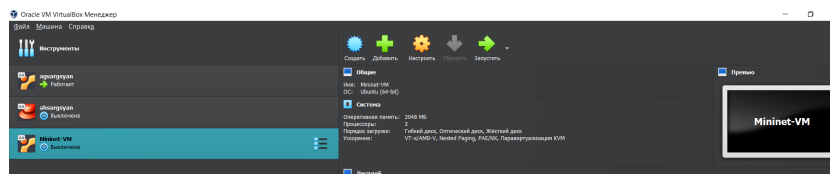


Рис. 2.1: установка mininet

```
Microsoft Windows [Version 10.0.19045.3448]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\WINDOWS\system32>choco install putty
Chocolatey v1.2.1
Installing the following packages:
putty
By installing, you accept licenses for the packages.
putty v0.79.0 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- putty - putty v0.79.0 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

C:\WINDOWS\system32>choco install vcxsrv
Chocolatey v1.2.1
Installing the following packages:
vcxsrv
By installing, you accept licenses for the packages.
vcxsrv v1.20.14.0 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- vcxsrv - vcxsrv v1.20.14.0 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

C:\WINDOWS\system32>
```

Рис. 2.2: установка putty и xserver

2. После запуск виртуальной машины и xserver, соединился с помощью консоли putty по ssh с ip адрессом 192.168.56.101

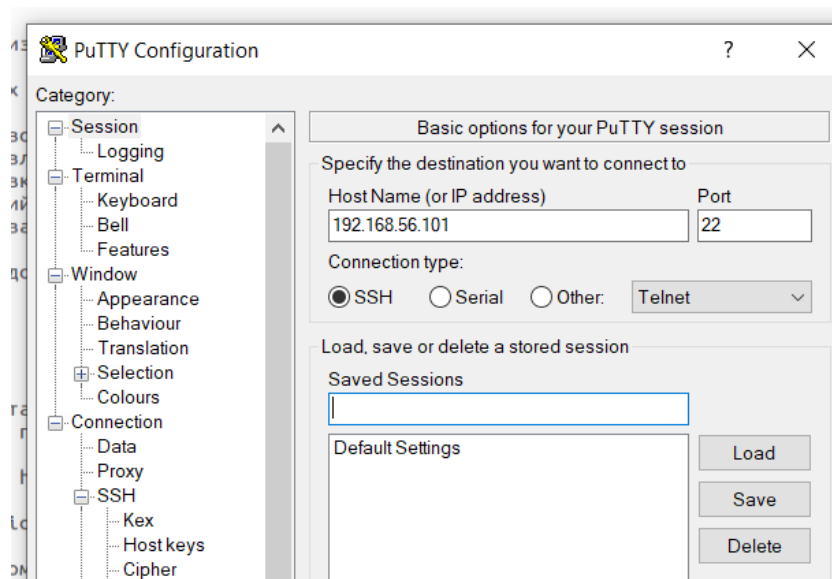


Рис. 2.3: подключение по ssh

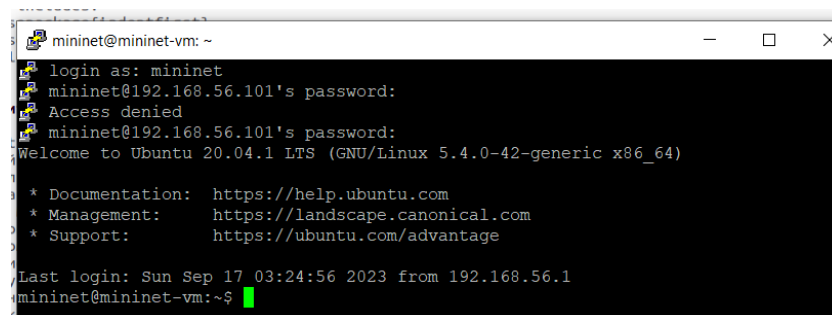


Рис. 2.4: виртуальная машина mininet

3. Настроил соединения X11 для суперпользователя



```
root@mininet-vm: ~
login as: mininet
mininet@192.168.56.101's password:
Access denied
mininet@192.168.56.101's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Last login: Sun Sep 17 03:24:56 2023 from 192.168.56.1
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 c93c676d5292566850caf9ef87513191
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 c93c676d52
2566850caf9ef87513191
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 c93c676d5292566850caf9ef87513191
root@mininet-vm:~#
```

Рис. 2.5: Настройка соединения X11 для суперпользователя

### 3 Процесс моделирования

1. Изучил основные команды mininet, в miniedit запустил простую топологию с двумя хостами и одним коммутатором

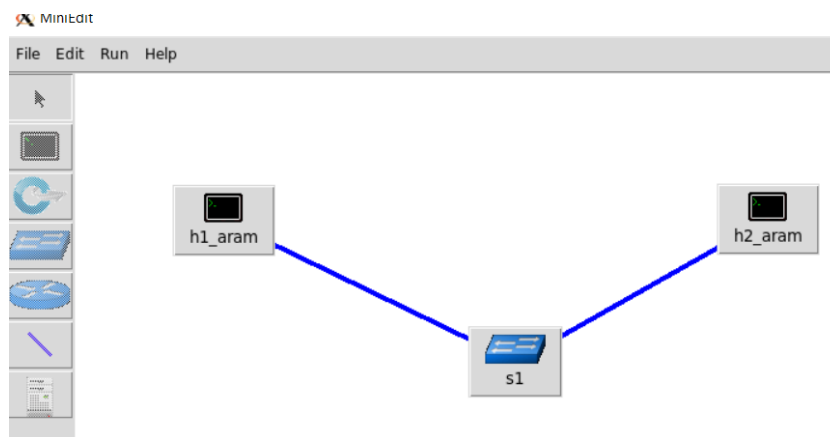


Рис. 3.1: Первая модель

2. Проверил работаспособность соединения

```

root@mininet-virtual-machine:/home/mininet# 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=0.916 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.085 ms
^C
--- 10.0.0.2 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10542ms
rtt min/avg/max/mdev = 0.047/0.154/0.916/0.241 ms
root@mininet-virtual-machine:/home/mininet#

```

Рис. 3.2: ping от первого хоста ко второму

### 3. Установил приложение iperf3

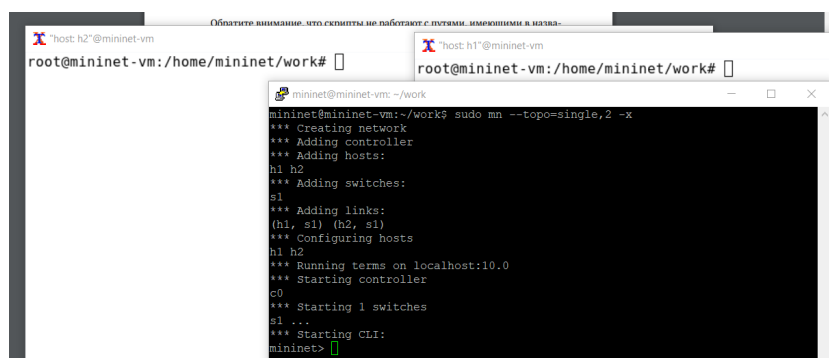
```

mininet@mininet-virtual-machine:~/work$ sudo apt install iperf3
Reading package lists... Done
Building dependency tree
Reading state information... Done
iperf3 is already the newest version (3.7-3).
0 upgraded, 0 newly installed, 0 to remove and 360 not upgraded.
mininet@mininet-virtual-machine:~/work$

```

Рис. 3.3: Установка iperf3

### 4. Смоделировал простую топологию с двумя хостами и коммутатором



```

root@mininet-virtual-machine:/home/mininet/work#
mininet@mininet-virtual-machine:~/work$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>

```

Рис. 3.4: Запуск модели

5. Провел простейший интерактивный эксперимент по измерению пропускной способности в `iperf3` и в терминале `mininet`. Как мы видим, при первом тесте потери пакетов значительно меньше (9 против 152), пропускная способность тоже больше в первом случае, как и размер окна перегрузки.

```
root@mininet-vm:/home/mininet# iperf3 -s
warning: this system does not seem to support IPv6 - trying IPv4
Server listening on 5201
Accepted connection from 10.0.0.1, port 55878
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 55880
[ ID] Interval      Transfer    Bitrate
[ 7] 0.00-1.00 sec  3.40 GBytes 29.2 Gbits/sec
[ 7] 1.00-2.00 sec  3.19 GBytes 27.4 Gbits/sec
[ 7] 2.00-3.00 sec  3.17 GBytes 27.3 Gbits/sec
[ 7] 3.00-4.00 sec  3.33 GBytes 28.6 Gbits/sec
[ 7] 4.00-5.00 sec  2.95 GBytes 25.3 Gbits/sec
[ 7] 5.00-6.00 sec  3.13 GBytes 26.9 Gbits/sec
[ 7] 6.00-7.00 sec  2.52 GBytes 21.7 Gbits/sec
[ 7] 7.00-8.00 sec  3.28 GBytes 28.2 Gbits/sec
[ 7] 8.00-9.00 sec  2.39 GBytes 20.5 Gbits/sec
[ 7] 9.00-10.00 sec 2.97 GBytes 25.5 Gbits/sec
[ ID] Interval      Transfer    Bitrate
[ 7] 0.00-10.00 sec 30.3 GBytes 26.1 Gbits/sec
Server listening on 5201

root@mininet-vm:/home/mininet# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 55880 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate      Retr    Cwnd
[ 7] 0.00-1.00 sec  3.40 GBytes 29.2 Gbits/sec    9   15.3 MBytes
[ 7] 1.00-2.00 sec  3.19 GBytes 27.4 Gbits/sec    0   15.3 MBytes
[ 7] 2.00-3.00 sec  3.18 GBytes 27.3 Gbits/sec    0   15.3 MBytes
[ 7] 3.00-4.00 sec  3.33 GBytes 28.6 Gbits/sec    0   15.3 MBytes
[ 7] 4.00-5.00 sec  2.95 GBytes 25.3 Gbits/sec    0   15.3 MBytes
[ 7] 5.00-6.00 sec  3.13 GBytes 26.9 Gbits/sec    0   15.3 MBytes
[ 7] 6.00-7.00 sec  2.52 GBytes 21.7 Gbits/sec    0   15.3 MBytes
[ 7] 7.00-8.00 sec  3.29 GBytes 28.2 Gbits/sec    0   15.3 MBytes
[ 7] 8.00-9.00 sec  2.39 GBytes 20.5 Gbits/sec    0   15.3 MBytes
[ 7] 9.00-10.00 sec 2.97 GBytes 25.5 Gbits/sec    0   15.3 MBytes
[ ID] Interval      Transfer    Bitrate      Retr
[ 7] 0.00-10.00 sec 30.3 GBytes 26.1 Gbits/sec    9
[ 7] 0.00-10.00 sec 30.3 GBytes 26.1 Gbits/sec    9
sender
receiver

iperf Done.
root@mininet-vm:/home/mininet#
```

Рис. 3.5: Первый тест

```
s1 ...
*** Starting CLI:
mininet> h2 iperf3 -s &
mininet> h1 iperf3 -c h2
Connecting to host 10.0.0.2, port 5201
[ 5] local 10.0.0.1 port 55884 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate      Retr    Cwnd
[ 5] 0.00-1.00 sec  2.46 GBytes 21.1 Gbits/sec    0   1.58 MBytes
[ 5] 1.00-2.00 sec  3.09 GBytes 26.6 Gbits/sec  152   1.66 MBytes
[ 5] 2.00-3.00 sec  3.45 GBytes 29.6 Gbits/sec    0   1.66 MBytes
[ 5] 3.00-4.00 sec  2.98 GBytes 25.5 Gbits/sec    0   1.66 MBytes
[ 5] 4.00-5.00 sec  2.89 GBytes 24.9 Gbits/sec    0   1.66 MBytes
[ 5] 5.00-6.00 sec  2.92 GBytes 25.1 Gbits/sec    0   1.66 MBytes
[ 5] 6.00-7.00 sec  2.84 GBytes 24.4 Gbits/sec    0   2.63 MBytes
[ 5] 7.00-8.00 sec  3.31 GBytes 28.4 Gbits/sec    0   2.63 MBytes
[ 5] 8.00-9.00 sec  3.06 GBytes 26.3 Gbits/sec    1   2.63 MBytes
[ 5] 9.00-10.00 sec 2.25 GBytes 19.3 Gbits/sec    0   2.63 MBytes
[ ID] Interval      Transfer    Bitrate      Retr
[ 5] 0.00-10.00 sec 29.3 GBytes 25.1 Gbits/sec  153
[ 5] 0.00-10.00 sec 29.3 GBytes 25.1 Gbits/sec  153
sender
receiver

iperf Done.
mininet>
```

Рис. 3.6: Второй тест

6. Провёл другой тест и визуализировал данные



Рис. 3.7: Окно перегрузки

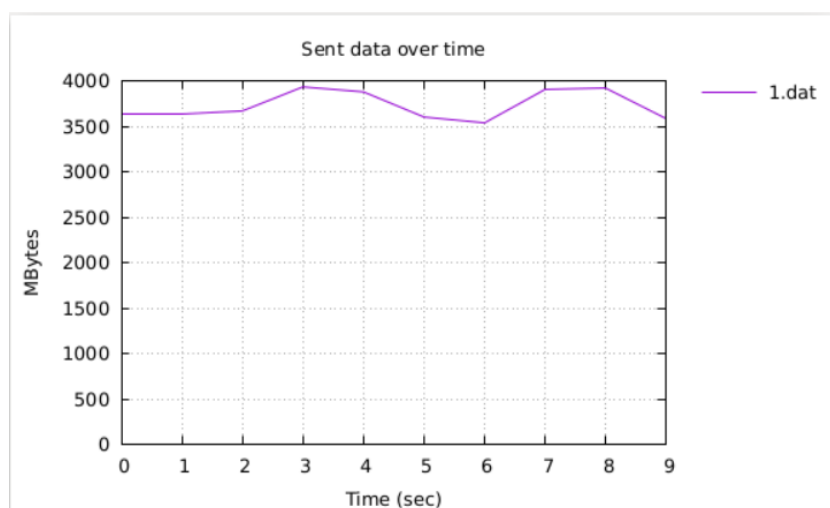


Рис. 3.8: Количество переданных байтов

7. С помощью API Mininet создал простейшую топологию сети, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8 и получил результаты

```

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

```

Рис. 3.9: Топология в API Mininet

```

mininet@mininet-vm:~$ mc
mininet@mininet-vm:~$
mininet@mininet-vm:~$ mc
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 a2:61:79:7f:48:fa
HOST h2 has IP address 10.0.0.2 and MAC address 16:fd:f8:85:e6:c6
*** Running CLI
*** Starting CLI:
mininet>

```

Рис. 3.10: Результаты