

Лабораторная работа № 1

Введение в Mininet

Шияпова Д.И.

20 мая 2025

Российский университет дружбы народов, Москва, Россия

- Шияпова Дарина Илдаровна
- Студентка
- Российский университет дружбы народов
- 1132226458@pfur.ru



Основной целью работы является развёртывание в системе виртуализации (например, в VirtualBox) mininet, знакомство с основными командами для работы с Mininet через командную строку и через графический интерфейс.

Mininet(**mininet?**) – это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины(VM). На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(`ifconfig`, `ping`) и даже получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик.

Mininet создает реалистичную виртуальную сеть, выполняя реальный код ядра, коммутатора и приложения на одной машине (VM, облачной или собственной) за считанные секунды с помощью одной команды `sudo mn`.

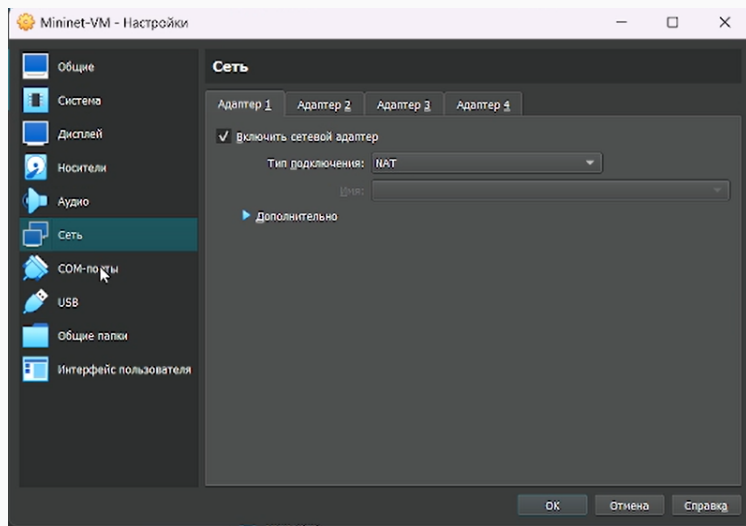


Рис. 1: Настройка сети

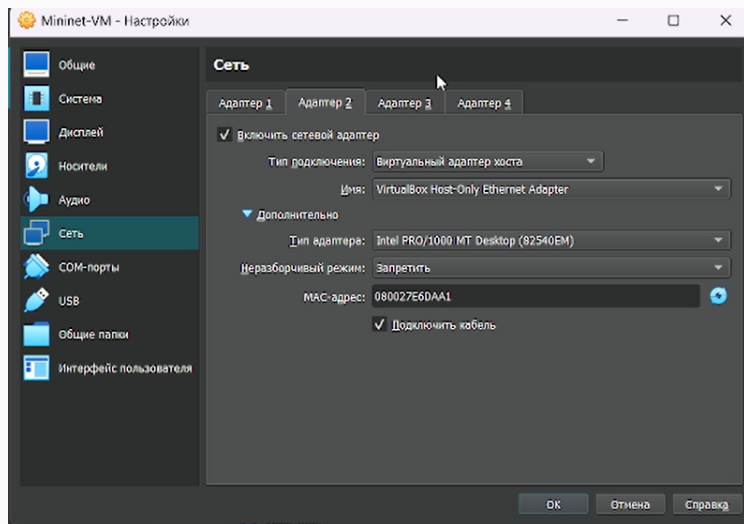
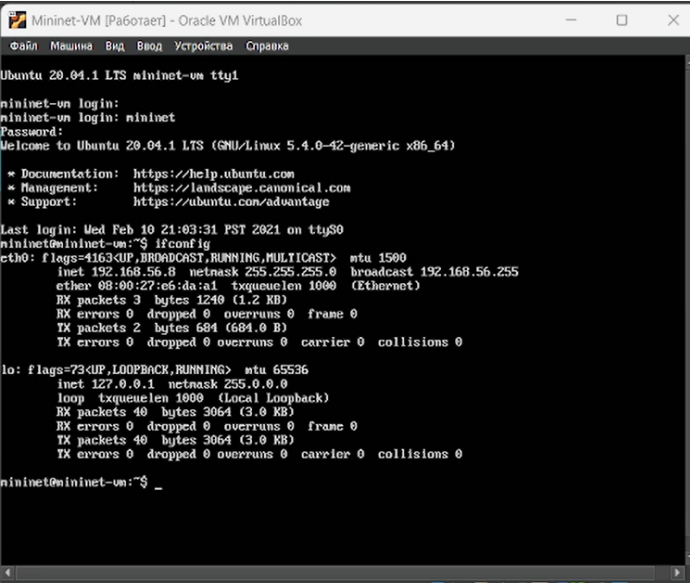


Рис. 2: Настройка сети

Выполнение лабораторной работы



```
Mininet-VM [Работаer] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

Ubuntu 20.04.1 LTS mininet-vm tty1

mininet-vm login:
mininet-vm login: mininet
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: Wed Feb 10 21:03:31 PST 2021 on ttyS0
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.8  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:c6:da:a1  txqueuelen 1000  (Ethernet)
    RX packets 3  bytes 1240 (1.2 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2  bytes 684 (684.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
    RX packets 40  bytes 3064 (3.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 40  bytes 3064 (3.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$ _
```

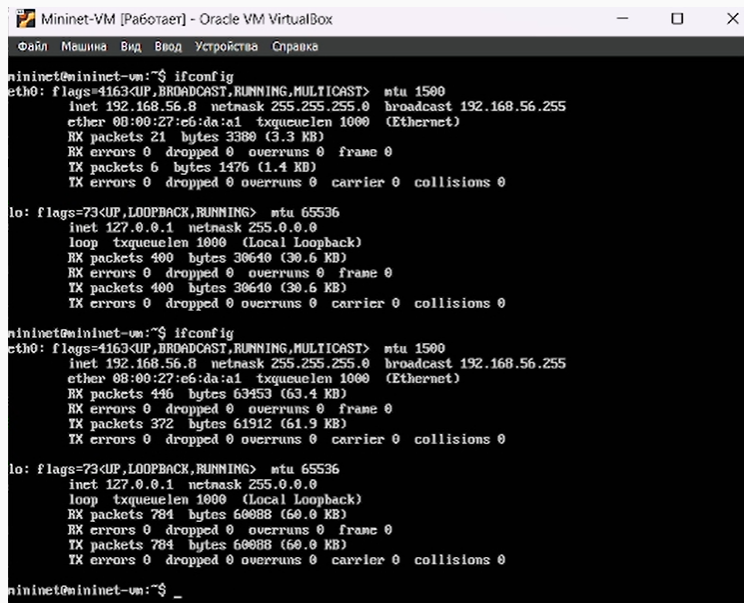
```
/usr/bin/xauth: file /home/mininet/.Xauthority does not exist
mininet@mininet-vm:~$
mininet@mininet-vm:~$ logout
Connection to 192.168.56.8 closed.
darina@LAPTOP-ONSDH9GT:~$ ssh -Y mininet@192.168.56.8
mininet@192.168.56.8's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
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: Wed Sep 10 11:01:32 2025 from 192.168.56.1
mininet@mininet-vm:~$ ssh-copy-id mininet@192.168.56.8
/usr/bin/ssh-copy-id: ERROR: No identities found
mininet@mininet-vm:~$ logout
Connection to 192.168.56.8 closed.
darina@LAPTOP-ONSDH9GT:~$ ssh-copy-id mininet@192.168.56.8
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/darina/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
mininet@192.168.56.8's password:
```

Рис. 4: Подключение к mininet через SSH

Выполнение лабораторной работы



```
Mininet-VM [Работаer] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.8  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:e6:da:a1  txqueuelen 1000  (Ethernet)
    RX packets 21  bytes 3380 (3.3 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 6  bytes 1476 (1.4 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
    RX packets 400  bytes 30640 (30.6 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 400  bytes 30640 (30.6 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.8  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:e6:da:a1  txqueuelen 1000  (Ethernet)
    RX packets 446  bytes 63453 (63.4 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 372  bytes 61912 (61.9 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
    RX packets 784  bytes 60088 (60.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 784  bytes 60088 (60.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$ _
```

```
/etc/netplan/01-netcfg.yaml  I-M--1 18 L:1 1+ 9 10/ 101 *(Z14 / Z14b) <EOF>  I*(11X1)
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: yes
    eth1:
      dhcp4: yes
```

Рис. 6: Файл /etc/netplan/01-netcfg.yaml

```
mininet@mininet-vm:~$  
mininet@mininet-vm:~$  
mininet@mininet-vm:~$ mv ~/mininet ~/mininet.orig  
mininet@mininet-vm:~$ cd ~  
mininet@mininet-vm:~$ git clone https://github.com/mininet/mininet.git  
Cloning into 'mininet'...  
remote: Enumerating objects: 10388, done.  
remote: Counting objects: 100% (128/128), done.  
remote: Compressing objects: 100% (60/60), done.  
remote: Total 10388 (delta 102), reused 60 (delta 60), pack-reused 10260 (from 3)  
Receiving objects: 100% (10388/10388), 3.36 MiB | 2.52 MiB/s, done.  
Resolving deltas: 100% (6905/6905), done.  
mininet@mininet-vm:~$ cd ~/mininet  
mininet@mininet-vm:~/mininet$ sudo make install  
cc -Wall -Wextra \\\n-DVERSION=\"PYTHONPATH=. python -B bin/mn --version 2>&1\" mnexec.c -o mnexec  
install -D mnexec /usr/bin/mnexec  
PYTHONPATH=. help2man -N -n "create a Mininet network." \\\n--no-discard-stderr "python -B bin/mn" -o mn.1  
help2man -N -n "execution utility for Mininet." \\\n-h "-h" -v "-v" --no-discard-stderr ./mnexec -o mnexec.1  
install -D -t /usr/share/man/man1 mn.1 mnexec.1  
python -m pip uninstall -y mininet || true  
Found existing installation: mininet 2.3.0  
Uninstalling mininet-2.3.0:  
  Successfully uninstalled mininet-2.3.0  
python -m pip install .  
Processing /home/mininet/mininet  
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from mininet==2.3.1b4)  
(45.2.0)  
Building wheels for collected packages: mininet  
-
```

```
mininet@mininet-vm:~/mininet$ mn --version  
2.3.1b4  
mininet@mininet-vm:~/mininet$ sudo _
```

Рис. 8: Номер установленной версии mininet

```
mininet@mininet-vm:~/mininet$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 667b953b15572667cd602272875a2bb5
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 b94ad20bf2c8f7655250d0e6df0c6ba5
mininet@mininet-vm:~/mininet$ MIT-MAGIC-COOKIE-1 667b953b15572667cd602272875a2bb5
MIT-MAGIC-COOKIE-1: command not found
mininet@mininet-vm:~/mininet$ /unix:10 MIT-MAGIC-COOKIE-1 667b953b15572667cd602272875a2bb5
-bash: /unix:10: No such file or directory
mininet@mininet-vm:~/mininet$ /unix:10 MIT-MAGIC-COOKIE-1 667b953b15572667cd602272875a2bb5
-bash: /unix:10: No such file or directory
mininet@mininet-vm:~/mininet$ unix:10 MIT-MAGIC-COOKIE-1 667b953b15572667cd602272875a2bb5
unix:10: command not found
mininet@mininet-vm:~/mininet$ MIT-MAGIC-COOKIE-1 667b953b15572667cd602272875a2bb5
MIT-MAGIC-COOKIE-1: command not found
mininet@mininet-vm:~/mininet$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 667b953b15572667cd602272875a2bb5
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 b94ad20bf2c8f7655250d0e6df0c6ba5
mininet@mininet-vm:~/mininet$ sudo -i
root@mininet-vm:~# xauth list
```

Рис. 9: Настройка соединения X11 для суперпользователя

Выполнение лабораторной работы

```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> help

Documented commands (type help <topic>):
=====
EOF      gterm  iperfudp  nodes      pingpair    py      switch  xterm
dpctl    help   link      noecho     pingpairfull  quit    time
dump     intfs  links     pingall    ports       sh      wait
exit     iperf  net       pingallfull  px          source  x

You may also send a command to a node using:
<node> command {args}
For example:
mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally b
mininet> xterm h2
```

```
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 36:3c:cb:6f:6c:97 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 11: Работа с Mininet с помощью командной строки

```
mininet> h2 ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 46:a0:db:37:7c:c6 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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=13.4 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.200 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.0
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.0
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.0
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.0
^C
```



```
--- 10.0.0.2 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8160ms
rtt min/avg/max/mdev = 0.043/1.554/13.427/4.197 ms
mininet> h2 ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=4.09 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.067 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.051 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.051 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.170 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.068 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.060 ms
^C
```

Рис. 13: Проверка связности хостов

```
mininet@mininet-vm:~$ sudo ~/mininet/mininet/examples/miniedit.py
Traceback (most recent call last):
  File "/home/mininet/mininet/mininet/examples/miniedit.py", line 3595, in <module>
    app = MiniEdit()
  File "/home/mininet/mininet/mininet/examples/miniedit.py", line 1123, in __init__
    Frame.__init__( self, parent )
  File "/usr/lib/python3.8/tkinter/__init__.py", line 3119, in __init__
    Widget.__init__(self, master, 'frame', cnf, (), extra)
  File "/usr/lib/python3.8/tkinter/__init__.py", line 2561, in __init__
    BaseWidget._setup(self, master, cnf)
  File "/usr/lib/python3.8/tkinter/__init__.py", line 2527, in _setup
    _default_root = Tk()
  File "/usr/lib/python3.8/tkinter/__init__.py", line 2261, in __init__
    self.tk = _tkinter.create(screenName, baseName, className, interactive, wantobjects, useTk, sync
, use)
_tkinter.TclError: no display name and no $DISPLAY environment variable
mininet@mininet-vm:~$
```

Рис. 14: sudo ~/mininet/mininet/examples/miniedit.py

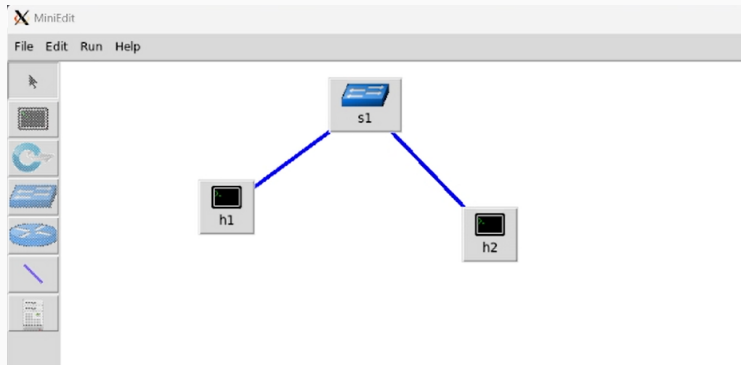


Рис. 15: Назначение IP-адресов

В результате выполнения данной лабораторной работы я развёрнула mininet в системе виртуализации VirtualBox, а также ознакомилась с основными командами для работы с Mininet через командную строку и через графический интерфейс.