

# **Введение в Mininet**

**Лабораторная работа № 1**

**Шулуужук Айраана НПИбд-02-22**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Подключение к виртуальной машине . . . . .	6
2.2	Настройка доступа к Интернет . . . . .	8
2.3	Обновление версии Mininet . . . . .	9
2.4	Настройка параметров XTerm . . . . .	10
2.5	Настройка соединения X11 для суперпользователя . . . . .	11
2.6	Работа с Mininet из-под Windows . . . . .	12
2.7	Основы работы в Mininet . . . . .	12
2.8	Построение и эмуляция сети в Mininet с использованием графического интерфейса . . . . .	16
<b>3</b>	<b>Выводы</b>	<b>19</b>

# Список иллюстраций

2.1	настройка виртуальной машины . . . . .	6
2.2	настройка сетевого адаптера . . . . .	7
2.3	запуск виртуальной машины и логин . . . . .	7
2.4	подключение к виртуальной машине из хостовой машины .	8
2.5	активирование NAT адреса . . . . .	8
2.6	установка mc . . . . .	9
2.7	редактирование файла . . . . .	9
2.8	обновлении версии . . . . .	10
2.9	проверка обновления . . . . .	10
2.10	установка шрифта . . . . .	11
2.11	настройка соединения для графических приложений . . . .	11
2.12	работа с Mininet из-под Windows . . . . .	12
2.13	запуск минимальной топологии . . . . .	13
2.14	отображение доступных узлов . . . . .	14
2.15	просмотр конфигурации узлов . . . . .	15
2.16	проверка связности . . . . .	15
2.17	запуск редактора MiniEdit и добавление топологии сети . . .	16
2.18	настройка IP адресации вручную . . . . .	17
2.19	проверка связности . . . . .	17
2.20	автоматическое назначение IP-адресов . . . . .	18
2.21	сохранение файла на хостовую машину . . . . .	18

## **Список таблиц**

# 1 Цель работы

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.

## 2 Выполнение лабораторной работы

### 2.1 Подключение к виртуальной машине

Запустим систему виртуализации и импортируем файл .ovf. Перейдем в настройки системы виртуализации и уточним параметры настройки виртуальной машины. Перейдите к опции «Система». Следуя рекомендациям, внесем исправления (рис. 2.1)

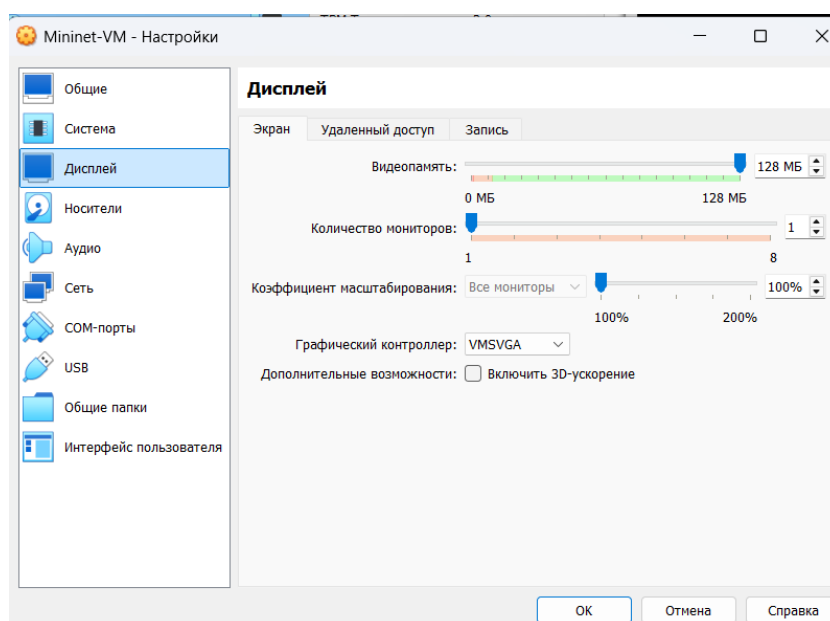


Рис. 2.1: настройка виртуальной машины

В настройках сети первый адаптер должен иметь подключение типа NAT. Для второго адаптера укажем тип подключения host-only network adapter (виртуальный адаптер хоста), который в дальнейшем мы будем

использовать для входа в образ виртуальной машины (рис. 2.2)

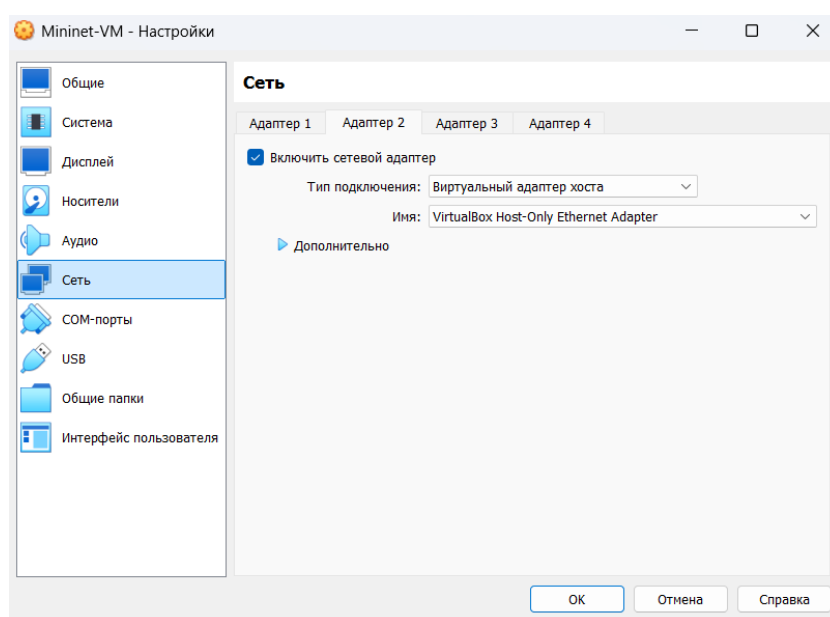


Рис. 2.2: настройка сетевого адаптера

Запустим виртуальную машину с Mininet (рис. 2.3)

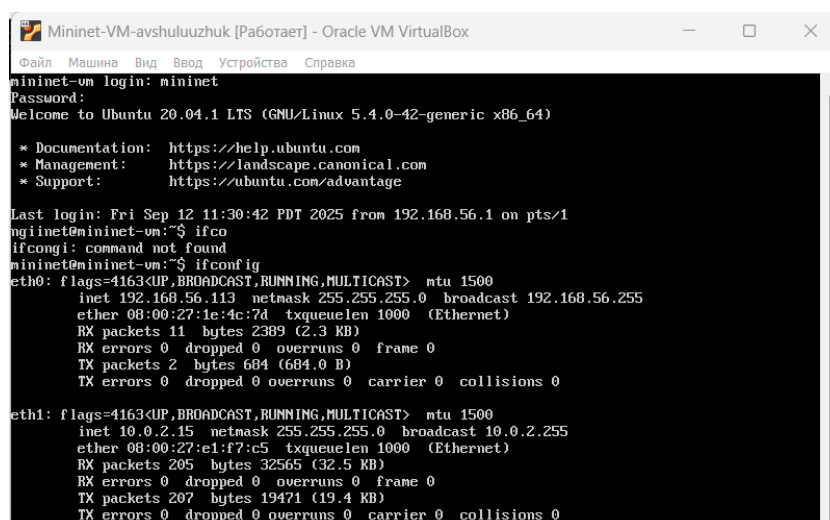


Рис. 2.3: запуск виртуальной машины и логин

Подключимся к виртуальной машине (из терминала хостовой машины) по ssh ключу. Подсоединение происходит успешно и без ввода пароля (рис. 2.4)

```
Last login: Fri Sep 12 09:14:18 2025 from 192.168.56.1
mininet@mininet-vm:~$ logout
Connection to 192.168.56.113 closed.

C:\Users\airan\.ssh>ssh -Y mininet@192.168.56.113
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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Sep 12 09:14:55 2025 from 192.168.56.1
mininet@mininet-vm:~$
```

Рис. 2.4: подключение к виртуальной машине из хостовой машины

## 2.2 Настройка доступа к Интернет

После подключения к виртуальной машине mininet посмотрим IP-адреса машины. Для доступа к сети Интернет должен быть активен адрес NAT: 10.0.0.x. Если активен только внутренний адрес машины вида 192.168.x.y, то активируем второй интерфейс (рис. 2.5)

```
mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.113 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:1e:4c:7d txqueuelen 1000 (Ethernet)
    RX packets 567 bytes 107470 (107.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 531 bytes 101562 (101.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    ether 08:00:27:e1:f7:c5 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 1180 (1.1 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 2000 bytes 153200 (153.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2000 bytes 153200 (153.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$
```

Рис. 2.5: активирование NAT адреса

Для удобства дальнейшей работы установим tc (рис. 2.6)



```

mininet@mininet-vm:~$ sudo apt install mc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libssh2-1 mc-data unzip
Suggested packages:
  arj catdvi | texlive-binaries dbview djvulibre-bin epub-utils genisoimage gv imagemagick
  libaspell-dev links | w3m | lynx odt2txt poppler-utils python python-boto python-tz xpdf
  | pdf-viewer zip
The following NEW packages will be installed:
  libssh2-1 mc mc-data unzip
0 upgraded, 4 newly installed, 0 to remove and 84 not upgraded.
Need to get 1,986 kB of archives.
After this operation, 8,587 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libssh2-1 amd64 1.8.0-2.1build1 [75.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 mc-data all 3:4.8.24-2ubuntu1 [1,265 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 mc amd64 3:4.8.24-2ubuntu1 [477 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal/main amd64 unzip amd64 6.0-25ubuntu1 [169 kB]
Fetched 1,986 kB in 3s (601 kB/s)
Selecting previously unselected package libssh2-1:amd64.
(Reading database ... 55%

```

Рис. 2.6: установка mc

Добавим для mininet указание на использование двух адаптеров при запуске. Для этого требуется перейти в режим суперпользователя и внести изменения в файл /etc/netplan/01-netcfg.yaml виртуальной машины mininet: (рис. 2.7)

```

mininet@mininet-vm: ~
/etc/netplan/01-netcfg.yaml [-M--] 11 L: [ 1+ 9 10/ 11] *(214 / 220b)
# 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. 2.7: редактирование файла

## 2.3 Обновление версии Mininet

В виртуальной машине mininet переименуем предыдущую установку Mininet. Скачаем новую версию Mininet и обновим исполняемые файлы (рис. 2.8).

```

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 68 (delta 68), pack-reused 10260 (from 3)
Receiving objects: 100% (10388/10388), 3.36 MiB | 885.00 KiB/s, done.
Resolving deltas: 100% (6985/6985), done.
mininet@mininet-vm:~$ cd ~/mininet
mininet@mininet-vm:~/mininet$ sudo male install
sudo: male: command not found
mininet@mininet-vm:~/mininet$ sudo make install
cc -Wall -Wextra \
-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." \
--no-discard-stderr "python -B bin/mn" -o mn.1
help2man -N -n "execution utility for Mininet." \
-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 .

```

Рис. 2.8: обновлении версии

Проверим номер установленной версии (рис. 2.9).

```

Building wheels for collected packages: mininet
  Building wheel for mininet (setup.py) ... done
  Created wheel for mininet: filename=mininet-2.3.1b4-py3-none-any.whl size=160942 sha256=bd99372
9688706e4185f2222a292b8c6402080be8d7164783af602cd36153
  Stored in directory: /tmp/pip-ephem-wheel-cache-zaq47vll/wheels/cd/7d/a7/aafe1b3eaff31efd6bafe2
c9698a717bdf739db6cfe8d45
Successfully built mininet
Installing collected packages: mininet
Successfully installed mininet-2.3.1b4
mininet@mininet-vm:~/mininet$ mn --version
2.3.1b4
mininet@mininet-vm:~/mininet$

```

Рис. 2.9: проверка обновления

## 2.4 Настройка параметров XTerm

По умолчанию XTerm использует растровые шрифты малого кегля. Для увеличения размера шрифта и применения векторных шрифтов вместо растровых необходимо внести изменения в файл /etc/X11/app-defaults/XTerm. Здесь выбран системный моноширинный шрифт, кегль шрифта — 12 пунктов (рис. 2.10).

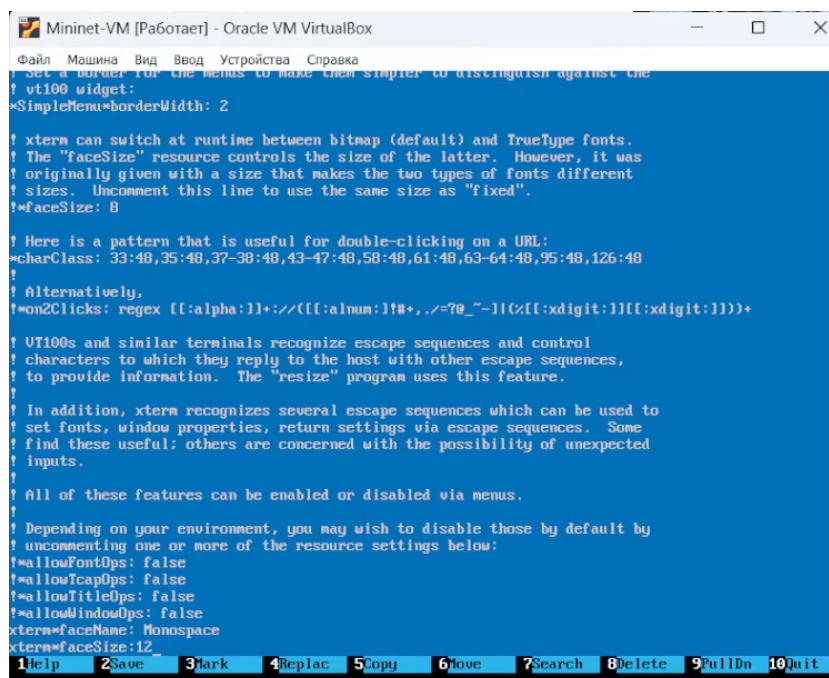


Рис. 2.10: установка шрифта

## 2.5 Настройка соединения X11 для суперпользователя

Скопируем значение куки (MIT magic cookie)<sup>1</sup> пользователя mininet в файл для пользователя root. После выполнения этих действий графические приложения должны запускаться под пользователем mininet (рис. 2.11)

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 583de15a3dd36497221f5d500990037f
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 01f7e6007504688ee72ac8d165dd613e
root@mininet-vm:~# xauth add mininet-vm/unix:11 ^C
root@mininet-vm:~# xauth add mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 583de15a3dd3
6497221f5d500990037f
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 583de15a3dd36497221f5d500990037f
root@mininet-vm:~# xterm
```

Рис. 2.11: настройка соединения для графических приложений

## 2.6 Работа с Mininet из-под Windows

Для работы с графическими приложениями я использовала Xserver: MobaXterm (рис. 2.12)

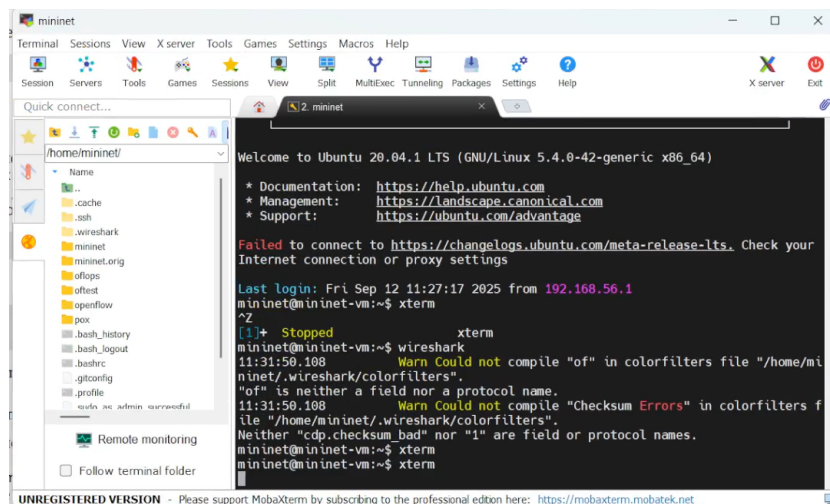


Рис. 2.12: работа с Mininet из-под Windows

## 2.7 Основы работы в Mininet

Вызов Mininet с использованием топологии по умолчанию. Для запуска минимальной топологии введем в командной строке (рис. 2.13)

```
mininet@mininet-vm:~/mininet$ cd
mininet@mininet-vm:~$ sudo mn
*** 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>
```

Рис. 2.13: запуск минимальной топологии

Для отображения списка команд интерфейса командной строки Mininet и примеров их использования введем команду в интерфейсе командной строки Mininet: `help`. Для отображения доступных узлов введите: `nodes`. Вывод этой команды показывает, что есть два хоста (хост h1 и хост h2) и коммутатор (s1) (рис. 2.14) (рис. 2.15)

```
mininet@mininet-vm: ~
*** 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 better:
mininet> xterm h2

mininet> nodes
available nodes are:
c0 h1 h2 s1
mininet> █ I
```

Рис. 2.14: отображение доступных узлов

Просмотрим доступные линки net. Вывод этой команды показывает: – Хост h1 подключён через свой сетевой интерфейс h1-eth0 к коммутатору на интерфейсе s1-eth1. – Хост h2 подключён через свой сетевой интерфейс h2-eth0 к коммутатору на интерфейсе s1-eth2. – Коммутатор s1: – имеет петлевой интерфейс lo. – подключается к h1-eth0 через интерфейс s1-eth1. – подключается к h2-eth0 через интерфейс s1-eth2. – Mininet позволяет выполнять команды на конкретном устройстве. Чтобы выполнить команду для определенного узла, необходимо сначала указать устройство, а затем команду, например: h1 ifconfig

Эта запись выполняет команду ifconfig на хосте h1 и показывает интерфейсы хоста h1 – хост h1 имеет интерфейс h1-eth0, настроенный с IP-адресом 10.0.0.1, и другой интерфейс lo, настроенный с IP-адресом 127.0.0.1. (рис. 2.15)

```

mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
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 ea:bc:b4:a8:99:46 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>

```

Рис. 2.15: просмотр конфигурации узлов

Чтобы проверить связь между ними, вы можете использовать команду `ping`. Команда `ping` работает, отправляя сообщения эхо-запроса протокола управляющих сообщений Интернета (ICMP) на удалённый компьютер и ожидая ответа. (рис. 2.16)

```

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=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.204 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.092 ms
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5079ms
rtt min/avg/max/mdev = 0.056/0.265/1.109/0.380 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 140.728 seconds
mininet@mininet-vm:~$

```

Рис. 2.16: проверка связности



## 2.8 Построение и эмуляция сети в Mininet с использованием графического интерфейса

В терминале виртуальной машины mininet запустим MiniEdit. Добавим два хоста и один коммутатор, соедините хосты с коммутатором (рис. 2.17)

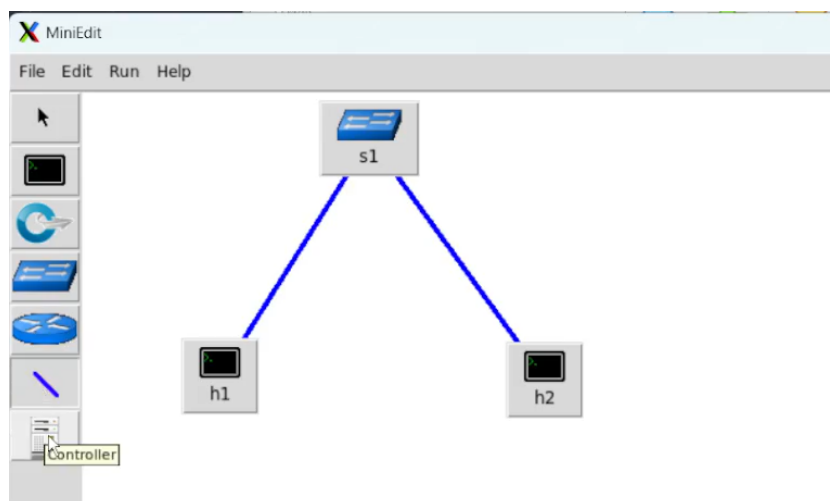


Рис. 2.17: запуск редактора MiniEdit и добавление топологии сети

Настроим IP-адреса на хостах h1 и h2. Для этого удерживая правую кнопку мыши на устройстве выберите свойства. Для хоста h1 укажем IP-адрес 10.0.0.1/8, а для хоста h2 — 10.0.0.2/8 (рис. 2.18)



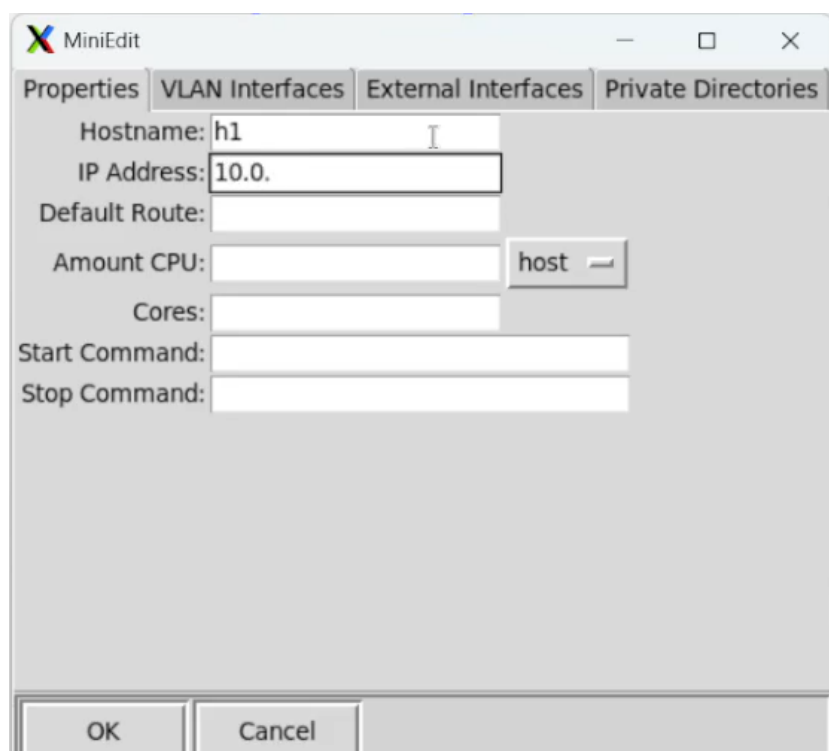


Рис. 2.18: настройка IP адресации вручную

Проверка связности. Пропингуем хосты h1 и h2 (рис. 2.19)

```

Host: h2"@"mininet-vm
root@mininet-vm:/home/mininet# 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=0.192 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=0.059 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.048 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.049 ms
^C
--- 10.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4100ms
rtt min/avg/max/mdev = 0.048/0.079/0.192/0.056 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.19: проверка связности

Автоматическое назначение IP-адресов. Ранее IP-адреса узлам h1 и h2 были назначены вручную. В качестве альтернативы можно полагаться на Mininet для автоматического назначения IP-адресов. По умолчанию в поле базовые значения IP-адресов (IP Base) установлено 10.0.0.0/8. Изменим это значение на 15.0.0.0/8 (рис. 2.20)

```
Host: h2@mininet-vm
Command 'ifconfog' not found, did you mean:
  command 'ifconfig' from deb net-tools (1.60+git20180626.aebd88e-1ubuntu1)
Try: apt install <deb name>

root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 15.0.0.2 netmask 255.0.0.0 broadcast 15.255.255.255
    ether 32:a2:50:dd:42:f4 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 955 bytes 247284 (247.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 955 bytes 247284 (247.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#
```

Рис. 2.20: автоматическое назначение IP-адресов

**Сохранение и загрузка топологии Mininet.** В домашнем каталоге виртуальной машины mininet создадим каталог для работы с проектами mininet. Укажем имя для топологии и сохраним на своём компьютере. После сохранения проекта поменяем права доступа к файлам в каталоге проекта (рис. 2.21)

```
PS C:\> cd Users
PS C:\Users> cd airan
PS C:\Users\airan> scp mininet@192.168.56.113:~/work/lab01.mn .
lab01.mn 100% 1654 807.6KB/s 00:00
PS C:\Users\airan> scp mininet@192.168.56.113:~/work/lab01.mn .
```

Рис. 2.21: сохранение файла на хостовую машину

## **3 Выводы**

В результате выполнения лабораторной работы было проведено развёртывание в системе виртуализации mininet, знакомство с основными командами для работы с Mininet через командную строку и через графический интерфейс.