

# Лабораторная работа №1 Введение в mininet

Леснухин Даниил Дмитриевич  
Российский университет дружбы народов  
Москва

## Содержание

<b>1</b>	<b>Цель работы</b>	<b>2</b>
<b>2</b>	<b>Задание</b>	<b>2</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>2</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>2</b>
4.1	Подключение к виртуальной машине . . . . .	4
4.2	Настройка доступа в интернет . . . . .	4
4.3	Обновление версии Mininet . . . . .	4
4.4	Настройка параметров XTerm . . . . .	4
4.5	Работа с Mininet из-под Windows . . . . .	6
4.6	Основы работы в Mininet . . . . .	8
4.7	Проверка связности . . . . .	8
4.8	Топология . . . . .	9
<b>5</b>	<b>Вывод</b>	<b>11</b>

## 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 Задание

Развернуть и настроить среду Mininet, изучить основы работы с эмулятором, построить простейшую топологию сети и проверить связность узлов.

## 3 Теоретическое введение

Mininet — это программная среда для эмуляции компьютерных сетей, позволяющая создавать виртуальные топологии с хостами, коммутаторами и каналами связи. В среде Mininet используются реальные сетевые приложения Linux и настоящий сетевой стек ядра, что делает результаты моделирования близкими к работе реальных сетей.

Mininet применяется для:

- тестирования сетевых протоколов,
- разработки и отладки сетевых приложений,
- моделирования архитектур программно-определяемых сетей (SDN),
- исследования пропускной способности и задержек.

Каждый виртуальный хост в Mininet представляет собой изолированное сетевое пространство имён (network namespace) Linux, имеющее собственные сетевые интерфейсы и IP-адреса. Коммутаторы эмулируются программно, чаще всего с использованием Open vSwitch.

По умолчанию Mininet позволяет запускать минимальную топологию, состоящую из одного коммутатора и двух хостов. Узлам автоматически назначаются IP-адреса из заданной подсети. Проверка связности осуществляется с использованием протокола ICMP и утилиты ping.

## 4 Выполнение лабораторной работы

В первую очередь нам необходимо установить и настроить нашу виртуальную машину с образом, который мы предварительно скачали с репозитория Mininet: <https://github.com/mininet/mininet>.

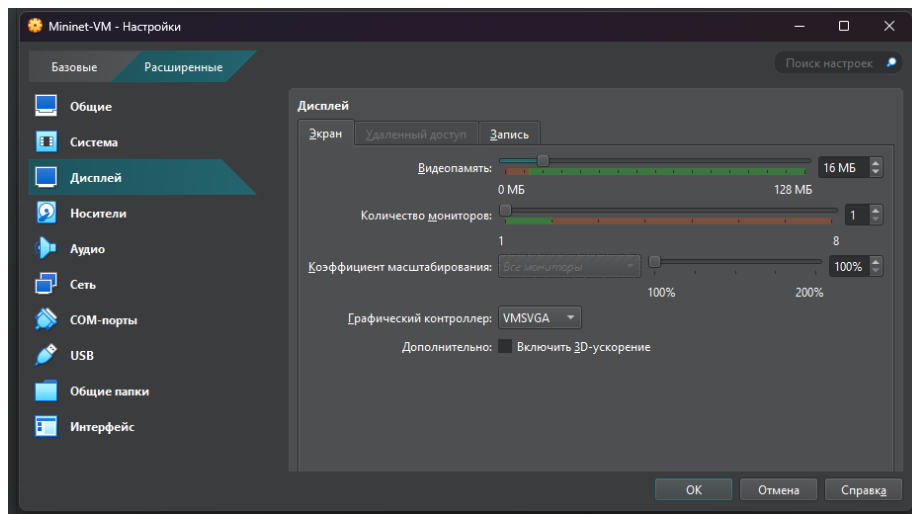


Рис. 1: Установка графического контроллера vmsvga

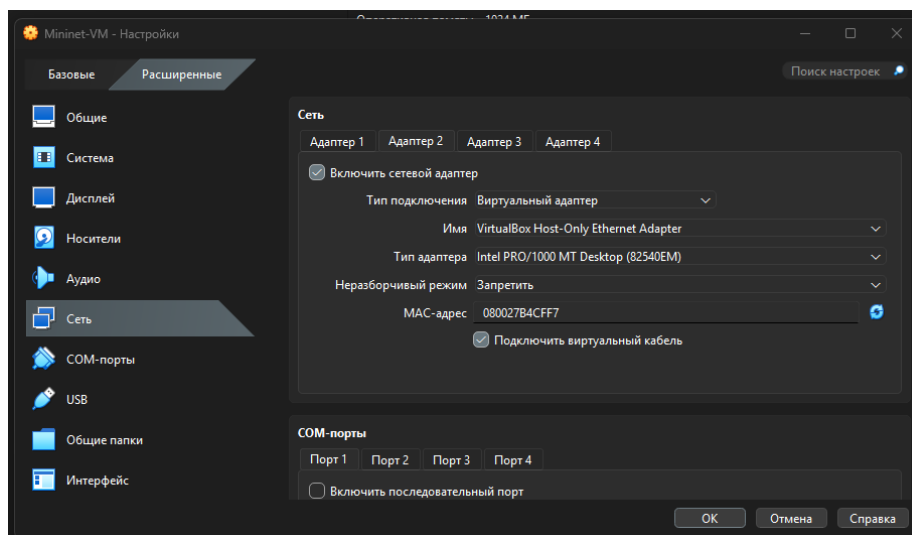
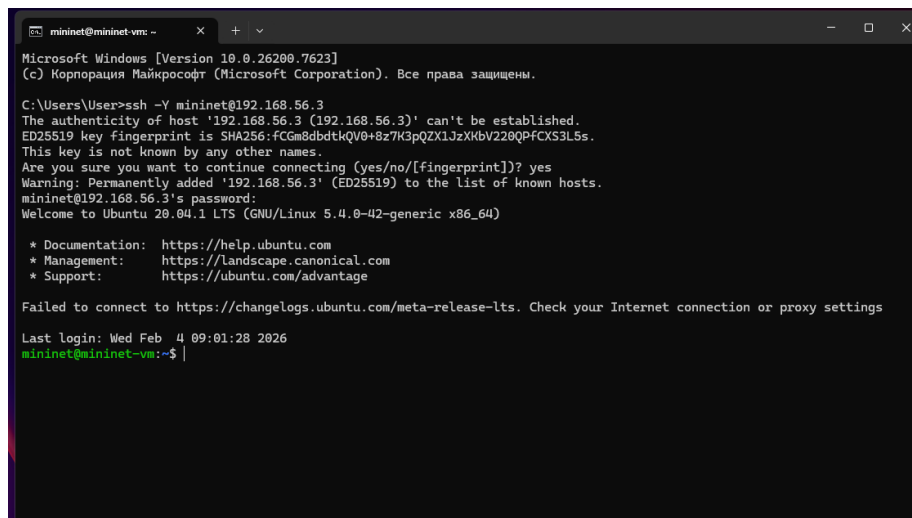


Рис. 2: Установка адаптера №2 в качестве виртуального

## 4.1 Подключение к виртуальной машине

Нам необходимо залогиниться в виртуальной машине под login: mininet и password: mininet и ввести команду ifconfig для получения внутреннего адреса машины



```
mininet@mininet-vm: ~  
Microsoft Windows [Version 10.0.26200.7623]  
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
  
C:\Users\User>ssh -Y mininet@192.168.56.3  
The authenticity of host '192.168.56.3 (192.168.56.3)' can't be established.  
ED25519 key fingerprint is SHA256:fGm8dbdtkQV0+8z7K3pQZK1JzXkbV220QPfCX53L5s.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.56.3' (ED25519) to the list of known hosts.  
mininet@192.168.56.3'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  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings  
  
Last login: Wed Feb  4 09:01:28 2026  
mininet@mininet-vm:~$
```

Рис. 3: Получение внутреннего адреса хоста

## 4.2 Настройка доступа в интернет

Для доступа к сети нам необходимо активировать второй интерфейс следующими командами

Теперь для нашего же дальнейшего удобства необходимо для mininet добавить указание на использование двух адаптеров при запуске. Для этого изменим файл /etc/netplan/01-netcfg.yaml

## 4.3 Обновление версии Mininet

– В виртуальной машине mininet переименуем предыдущую установку Mininet:  
mv ~/mininet ~/mininet.orig

– Скачаем новую версию Mininet: 1 cd ~ 2 git clone https://github.com/mininet/mininet.git  
cd ~ git clone https://github.com/mininet/mininet.git

Обновим исполняемые файлы: cd ~/mininet sudo make install

## 4.4 Настройка параметров XTerm

По умолчанию XTerm использует растровые шрифты малого кегля. Для увеличения размера шрифта и применения векторных шрифтов вместо растровых необ-

```

mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.3 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:27:42:c6 txqueuelen 1000 (Ethernet)
    RX packets 135 bytes 17744 (17.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 103 bytes 16704 (16.7 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 08:00:27:2a:b6:e7 txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 2137 (2.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7 bytes 1360 (1.3 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 284 bytes 21824 (21.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 284 bytes 21824 (21.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$

```

Рис. 4: Активация второго интерфейса

```

mininet@mininet-vm: ~
/etc/netplan/01-netcfg.yaml [-M--] 16 L: [ 1+ 9 10/ 10] *(219 / 219b) <EOF>
# 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. 5: изменение файла /etc/netplan/01-netcfg.yaml

```
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% (136/136), done.
remote: Compressing objects: 100% (64/64), done.
remote: Total 10388 (delta 109), reused 72 (delta 72), pack-reused 10252 (from 2)
Receiving objects: 100% (10388/10388), 3.36 MiB | 8.45 MiB/s, done.
Resolving deltas: 100% (6909/6909), done.
mininet@mininet-vm:~$ cd ~/mininet
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 .
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
  Building wheel for mininet (setup.py) ... done
  Created wheel for mininet: filename=mininet-2.3.1b4-py3-none-any.whl size=160942 sha256=053653b7925295bd88224c063a1e8b7fd021db89ae5133e660ade4bbfc88832
  Stored in directory: /tmp/pip-ephem-wheel-cache-bz_7aeda/wheels/cd/7d/a7/aafelb3eaff31efd6ba4e2ea6c9690a717bdf739db6cf
e8d45
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. 6: Обновление версии mininet

ходимо внести изменения в файл /etc/X11/app-defaults/XTerm. Для этого можно воспользоваться следующей командой:

```
sudo mcedit /etc/X11/app-defaults/XTerm
```

и затем в конце файла добавить строки: *xterm* *faceName: Monospace* *xterm* *faceSize: 12*

```
! Depending on your environment, you may wish to disable those by default by
! uncommenting one or more of the resource settings below:
!*allowFontOps: false
!*allowTcapOps: false
!*allowTitleOps: false
!*allowWindowOps: false

xterm*faceName: Monospace
xterm*faceSize: 12
1Help      2Save      3Mark      4Replac    5Copy      6Move      7Search    8Delete    9PullDn   10Quit
```

Рис. 7: Настройка параметров XTerm

## 4.5 Работа с Mininet из-под Windows

Установка программного обеспечения. – Установим putty: `choco install putty` –  
Установите VcXsrv Windows X Server: `choco install vcxsrv`

```
Администратор: Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> choco install putty
chocolatey v2.4.1
Installing the following packages:
putty
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'

putty.portable v0.83.0 [Approved]
putty.portable package files install completed. Performing other installation steps.
The package putty.portable wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y

Extracting 64-bit C:\ProgramData\chocolatey\lib\putty.portable\tools\putty_x64.zip to C:\ProgramData\chocolatey\lib\putty
y.portable\tools...
C:\ProgramData\chocolatey\lib\putty.portable\tools
ShimGen has successfully created a gui shim for PAGEANT.EXE
ShimGen has successfully created a shim for PLINK.EXE
ShimGen has successfully created a shim for PSCP.EXE
ShimGen has successfully created a shim for PSFTP.EXE
ShimGen has successfully created a gui shim for PUTTY.EXE
ShimGen has successfully created a gui shim for PUTTYGEN.EXE
The install of putty.portable was successful.
Deployed to 'C:\ProgramData\chocolatey\lib\putty.portable\tools'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading putty 0.83.0... 100%

putty v0.83.0 [Approved]
putty package files install completed. Performing other installation steps.
The install of putty was successful.
Deployed to 'C:\ProgramData\chocolatey\lib\putty'

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32> choco install vcxsrv
chocolatey v2.4.1
Installing the following packages:
vcxsrv
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading vcxsrv 21.1.10... 100%

vcxsrv v21.1.10 [Approved]
vcxsrv package files install completed. Performing other installation steps.
The package vcxsrv wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y

Installing 64-bit vcxsrv...
vcxsrv has been installed.
vcxsrv may be able to be automatically uninstalled.
The install of vcxsrv was successful.
Software installed as 'EXE', install location is likely default.

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32>
```

Рис. 8: Работа с Mininet из-под Windows

## 4.6 Основы работы в Mininet

Вызов Mininet с использованием топологии по умолчанию. — Для запуска минимальной топологии введите в командной строке:

```
sudo mn
```

```
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> help

Documented commands (type help <topic>):
=====
EOF      gterm  iperfudp  nodes      pingpair    py      switch  xterm
dptcl    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> |
```

Рис. 9: Основы работы в Mininet

## 4.7 Проверка связности

По умолчанию узлам h1 и h2 назначаются IP-адреса 10.0.0.1/8 и 10.0.0.2/8 соответственно. 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.



Например, команда

```
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=1.62 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.181 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.045 ms
^C
-- 10.0.0.2 ping statistics --
8 packets transmitted, 8 received, 0% packet loss, time 7153ms
rtt min/avg/max/mdev = 0.042/0.260/1.618/0.515 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
.
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 159.878 seconds
mininet@mininet-vm:~$
```

Рис. 10: Проверка связности

## 4.8 Топология

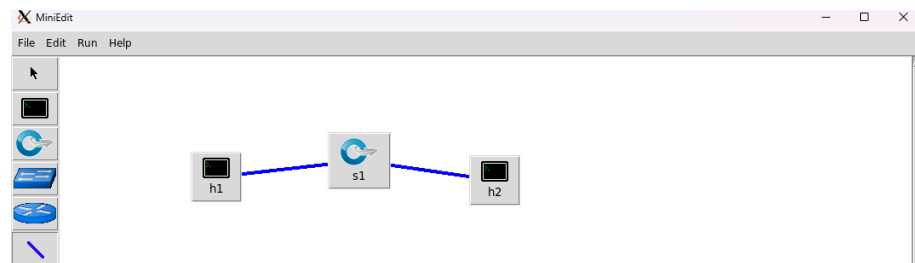


Рис. 11: строим топологию

Проверяем связь между двумя узлами

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. – Удалите назначенный вручную IP-адрес с хостов h1 и h2. – В MiniEdit нажмите Edit Preferences . По умолчанию в поле базовые значения IP-адресов (IP Base) установлено 10.0.0.0/8. Измените это значение на 15.0.0.0/8.

Проверка работы

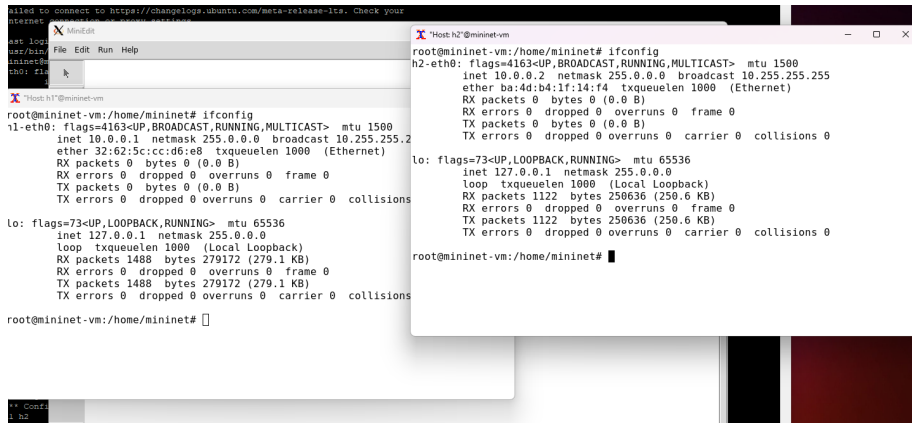


Рис. 12: проверка связи

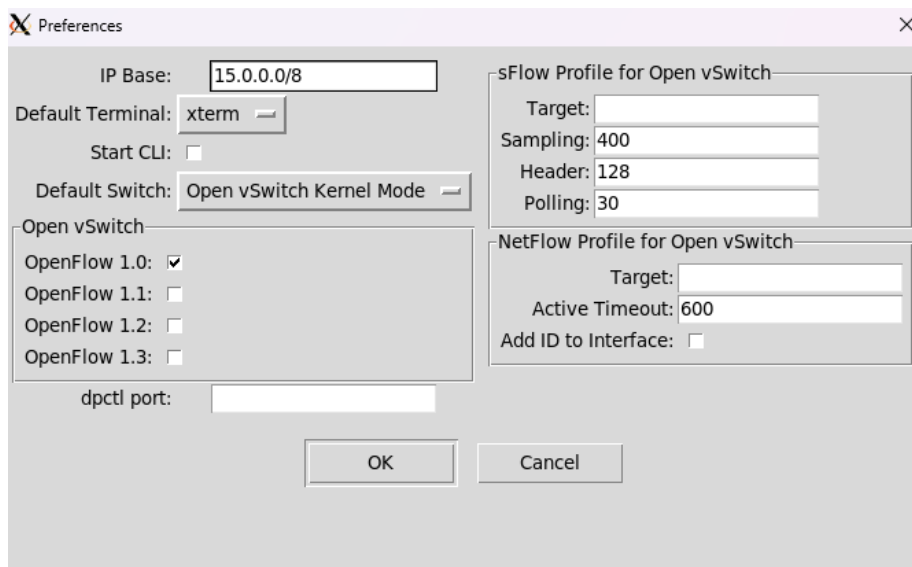


Рис. 13: Изменение ip - адреса вручную

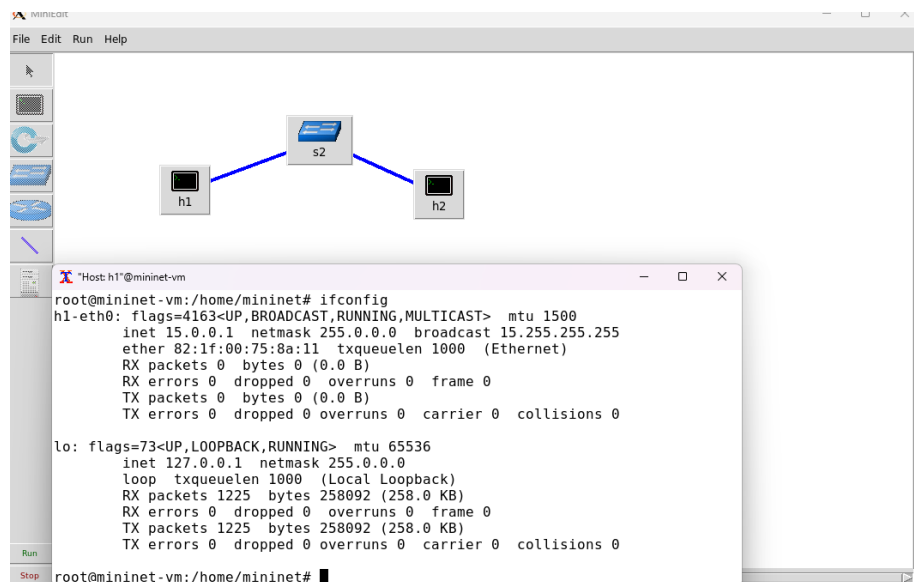


Рис. 14: Запуск топологии с новыми адресами

## 5 Вывод

В ходе лабораторной работы была развернута и настроена виртуальная среда Mininet в системе виртуализации VirtualBox. Были изучены основные принципы работы эмулятора сетей, выполнена настройка сетевых интерфейсов и организовано SSH-подключение к виртуальной машине.

В процессе работы была запущена минимальная топология, исследована её структура и определены IP-адреса узлов. Проверка связности между хостами с использованием утилиты ping показала корректную работу сети.

Также была построена аналогичная топология в графическом интерфейсе MiniEdit и изучен механизм автоматического назначения IP-адресов.

Таким образом, были получены практические навыки работы с Mininet и освоены базовые методы моделирования сетей передачи данных.