

# **Отчёт по лабораторной работе №1**

**Введение в Mininet**

Ким Реачна

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	20

# Список иллюстраций

2.1	Установка виртуальной машины Mininet . . . . .	5
2.2	Установка Putty и VcXsrv . . . . .	6
2.3	Конфигурация Xserver: Display setting . . . . .	6
2.4	Конфигурация Xserver: Start no client . . . . .	7
2.5	Дополнительная настройка: Disable access control . . . . .	7
2.6	Запуск Putty . . . . .	8
2.7	IP-адреса до активного NAT . . . . .	9
2.8	IP-адреса машины . . . . .	9
2.9	Установка mc . . . . .	10
2.10	Указание на использование двух адаптеров . . . . .	10
2.11	Обновление версии Mininet . . . . .	11
2.12	Изменения в файле /etc/X11/app-defaults/XTerm . . . . .	12
2.13	MIT magic cookie . . . . .	12
2.14	Основы работы в Mininet . . . . .	13
2.15	Информации о конфигурации h1 . . . . .	13
2.16	Проверка связности . . . . .	14
2.17	Завершения эмуляции . . . . .	14
2.18	Построение топологии . . . . .	15
2.19	IP-адрес для хоста h1 . . . . .	15
2.20	IP-адрес для хоста h2 . . . . .	16
2.21	Проверка связности ping . . . . .	17
2.22	Именование IP-адрес . . . . .	17
2.23	Отображение IP-адресов . . . . .	18
2.24	Сохранение топологии . . . . .	18
2.25	Право доступа к файлу . . . . .	19

# 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

1. Я установила необходимое оборудование для работы с Mininet из-под ОС Windows:

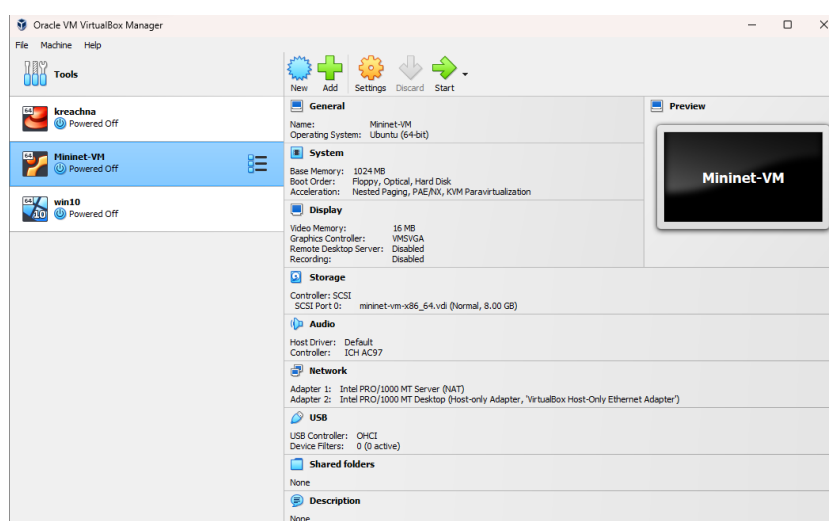


Рис. 2.1: Установка виртуальной машины Mininet

```
Administrator: Windows PowerShell
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> choco install Putty
chocolatey v1.1.0
2 validations performed. 1 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:
- A pending system reboot request has been detected, however, this is
  being ignored due to the current Chocolatey configuration. If you
  want to halt when this occurs, then either set the global feature
  using:
    choco feature enable -name-exitOnRebootDetected
  or pass the option --exit-when-reboot-detected.

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.

Did you know the proceeds of Pro (and some proceeds from other
licensed editions) go into bettering the community infrastructure?
Your support ensures an active community, keeps Chocolatey tip-top,
plus it nets you some awesome features!
https://chocolatey.org/compare
PS C:\WINDOWS\system32> choco install vcxsrv
chocolatey v1.1.0
2 validations performed. 1 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:
- A pending system reboot request has been detected, however, this is
  being ignored due to the current Chocolatey configuration. If you
  want to halt when this occurs, then either set the global feature
  using:
    choco feature enable -name-exitOnRebootDetected
  or pass the option --exit-when-reboot-detected.

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).
```

Рис. 2.2: Установка Putty и VcXsrv

## 2. Запустите Xserver и запуск Putty:

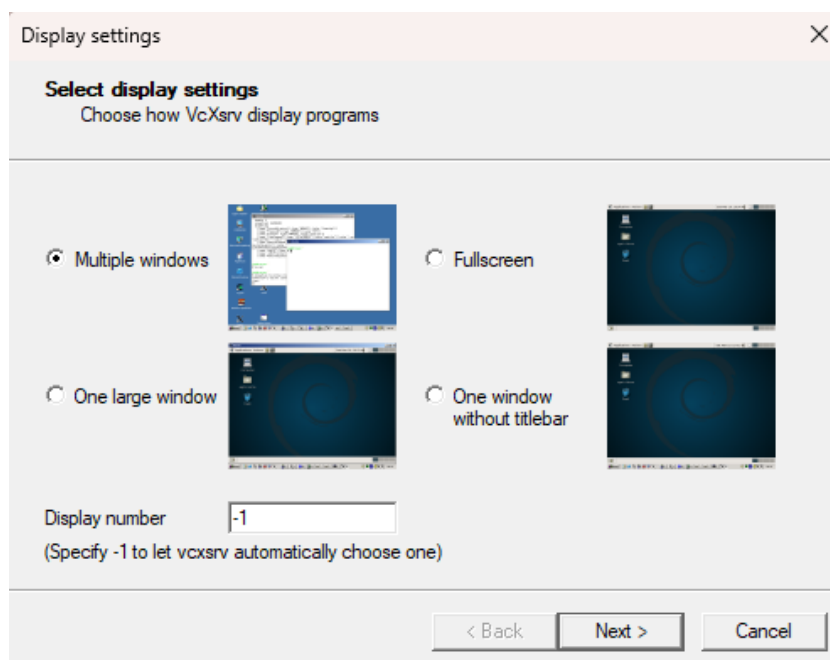


Рис. 2.3: Конфигурация Xserver: Display setting

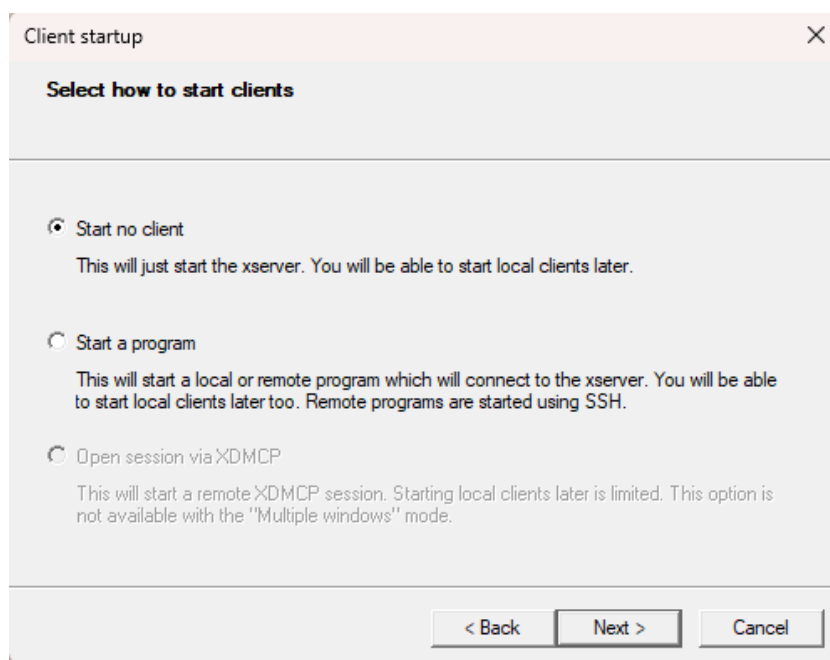


Рис. 2.4: Конфигурация Xserver: Start no client

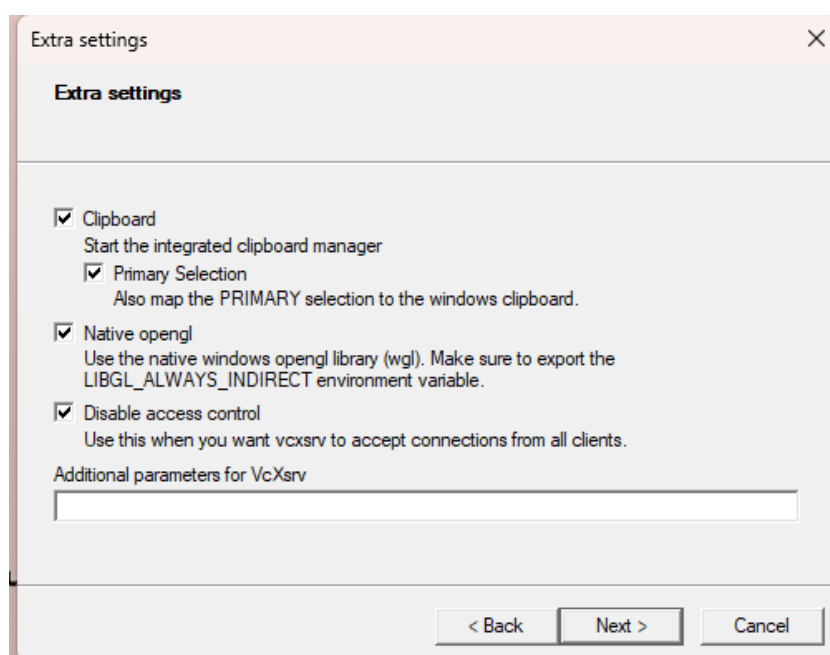


Рис. 2.5: Дополнительная настройка: Disable access control

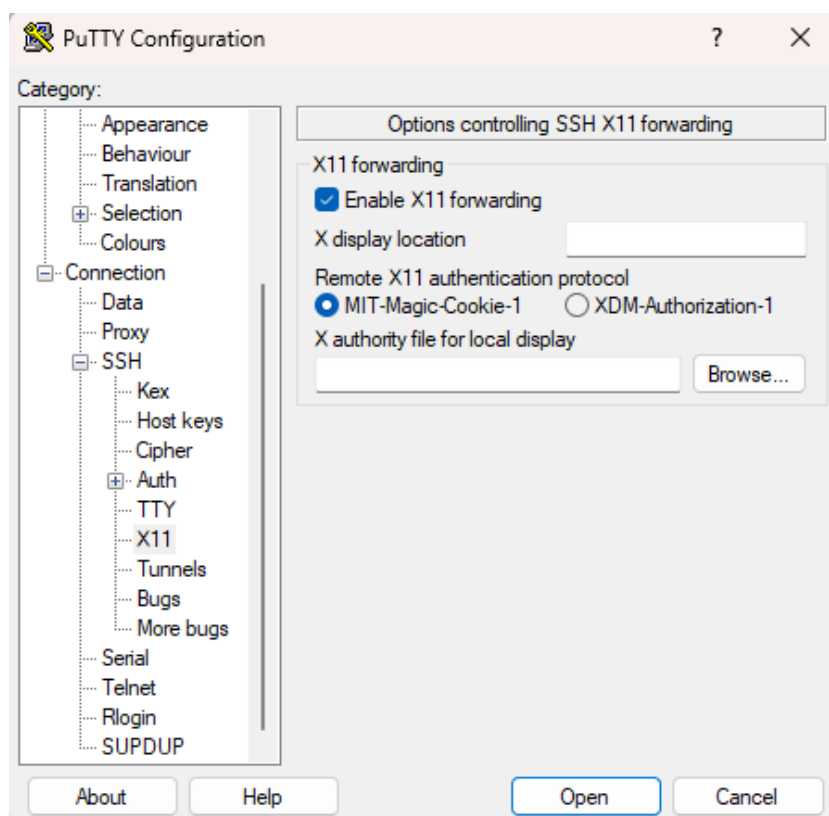


Рис. 2.6: Запуск Putty

- После подключения к виртуальной машине mininet посмотрите 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 активируйте второй интерфейс, набрав в командной строке:

```
sudo dhclient eth1  
ifconfig
```



```
mininet@mininet-vm: ~  
login as: mininet  
mininet@192.168.56.104'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: Thu Nov 16 04:35:29 2023 from 192.168.56.1  
mininet@mininet-vm:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500  
    inet 192.168.56.104  netmask 255.255.255.0  broadcast 192.168.56.255  
    ether 08:00:27:fd:6d:ca  txqueuelen 1000  (Ethernet)  
    RX packets 108  bytes 16607 (16.6 KB)  
    RX errors 0  dropped 0  overruns 0  frame 0  
    TX packets 82  bytes 14373 (14.3 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 320  bytes 24512 (24.5 KB)  
    RX errors 0  dropped 0  overruns 0  frame 0  
    TX packets 320  bytes 24512 (24.5 KB)  
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Рис. 2.7: IP-адреса до активного NAT

```
mininet@mininet-vm:~$ sudo dhclient eth1  
mininet@mininet-vm:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500  
    inet 192.168.56.104  netmask 255.255.255.0  broadcast 192.168.56.255  
    ether 08:00:27:fd:6d:ca  txqueuelen 1000  (Ethernet)  
    RX packets 166  bytes 21711 (21.7 KB)  
    RX errors 0  dropped 0  overruns 0  frame 0  
    TX packets 112  bytes 18719 (18.7 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 08:00:27:86:d6:24  txqueuelen 1000  (Ethernet)  
    RX packets 65  bytes 7261 (7.2 KB)  
    RX errors 0  dropped 0  overruns 0  frame 0  
    TX packets 67  bytes 6528 (6.5 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 343  bytes 26858 (26.8 KB)  
    RX errors 0  dropped 0  overruns 0  frame 0  
    TX packets 343  bytes 26858 (26.8 KB)  
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Рис. 2.8: IP-адреса машины

4. Для удобства дальнейшей работы установите mc: `sudo apt install mc`

```

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 dvi2ps 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 94 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 6.0-25ubu
ntul [169 kB]
Fetched 1,986 kB in 3s (755 kB/s)
Selecting previously unselected package libssh2-1:amd64.
(Reading database ... 101729 files and directories currently installed.)
Preparing to unpack .../libssh2-1_1.8.0-2.1build1_amd64.deb ...
Unpacking libssh2-1:amd64 (1.8.0-2.1build1) ...
Selecting previously unselected package mc-data.
Preparing to unpack .../mc-data_3%3a4.8.24-2ubuntu1_all.deb ...
Unpacking mc-data (3:4.8.24-2ubuntu1) ...
Selecting previously unselected package mc.

```

Рис. 2.9: Установка mc

5. Для удобства дальнейшей работы добавьте 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: `sudo mcedit /etc/netplan/01-netcfg.yaml`

```

mininet@mininet-vm: ~
/etc/netplan/01-netcfg.yaml [-M--] 6 L:[ 1+ 9 10/ 11] *(209 / 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. 2.10: Указание на использование двух адаптеров

6. Обновила версию mininet:
  - В виртуальной машине mininet переименуйте предыдущую установку Mininet: `mv ~/mininet ~/mininet.orig`

- Скачайте новую версию Mininet:

```
cd ~
```

```
git clone https://github.com/mininet/mininet.git
```

- Обновите исполняемые файлы:

```
cd ~/mininet
```

```
sudo make install
```

- Проверьте номер установленной версии mininet: `mn --version`

```
mininet@mininet-vm:~$ mv ~/mininet ~/mininet.orig
mininet@mininet-vm:~$ cd ~
mininet@mininet-vm:~$ mn --version
2.3.0
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% (234/234), done.
remote: Compressing objects: 100% (140/140), done.
remote: Total 10388 (delta 129), reused 175 (delta 92), pack-reused 10154
Receiving objects: 100% (10388/10388), 3.36 MiB | 4.30 MiB/s, done.
Resolving deltas: 100% (6910/6910), 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=0ef19e49b7467443838bbed790fbadbfffd706ed4d063e588ac186c36f2981ce
  Stored in directory: /tmp/pip-ephem-wheel-cache-194m7kjb/wheels/cd/7d/a7/aafef1b3eaff31efdb6a4e2ea6c9e90a717bdf739db6cfe8d45
Successfully built mininet
Installing collected packages: mininet
Successfully installed mininet-2.3.1b4
mininet@mininet-vm:~/mininet$ mn --version
2.3.1b4
```

Рис. 2.11: Обновление версии Mininet

7. Внесение изменений 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
!*allowTcpOps: false
!*allowTitleOps: false
!*allowWindowOps: false
xterm*faceName: Monospace
xterm*faceSize: 12
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис. 2.12: Изменения в файле /etc/X11/app-defaults/XTerm

8. Скопируйте значение куки (MIT magic cookie)1 пользователя mininet в файл для пользователя root:

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 4f04e38fa64e0dd23b1a77dcf3d90ed0
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
xauth: file /root/.Xauthority does not exist
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 4f04e38fa64e0dd23b1a77dcf3d90ed0
xauth: file /root/.Xauthority does not exist
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 4f04e38fa64e0dd23b1a77dcf3d90ed0
root@mininet-vm:~# logout
```

Рис. 2.13: MIT magic cookie

9. Вызов Mininet с использованием топологии по умолчанию:

- Для запуска минимальной топологии введите в командной строке: `sudo mn`
- Для отображения списка команд интерфейса командной строки Mininet и примеров их использования введите команду в интерфейсе командной строки Mininet: `help`
- Для отображения доступных узлов введите: `nodes`
- Введите команду `net` в интерфейсе командной строки Mininet, чтобы просмотреть доступные линки: `net`

```

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
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> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0

```

Рис. 2.14: Основы работы в Mininet

- Отображение информации о конфигурации сетевого интерфейса для хоста h1: h1 config

```

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 ce:1a:2:57:47:48:24 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

```

Рис. 2.15: Информации о конфигурации h1

10. Проверка соединения между хостами h1 и h2 использовать команду ping h1  
ping 10.0.0.2 и остановка эмуляции использовать команду exit

```

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=2.44 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.209 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.036 ms
^C
--- 10.0.0.2 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11238ms
rtt min/avg/max/mdev = 0.031/0.254/2.439/0.660 ms

```

Рис. 2.16: Проверка связности

```

mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 307.331 seconds
mininet@mininet-vm:~$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller
ovs-testcontroller udprawtest mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-control
ler ovs-testcontroller udprawtest mnexec ivs ryu-manager 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([_.:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.

```

Рис. 2.17: Завершения эмуляции

11. Построение топологии сети: Добавьте два хоста и один коммутатор и Настройте IP-адреса на хостах h1 и h2. Для хоста h1 укажите IP-адрес 10.0.0.1/8, а для хоста h2 — 10.0.0.2/8: `sudo ~/mininet/mininet/examples/miniedit.py`

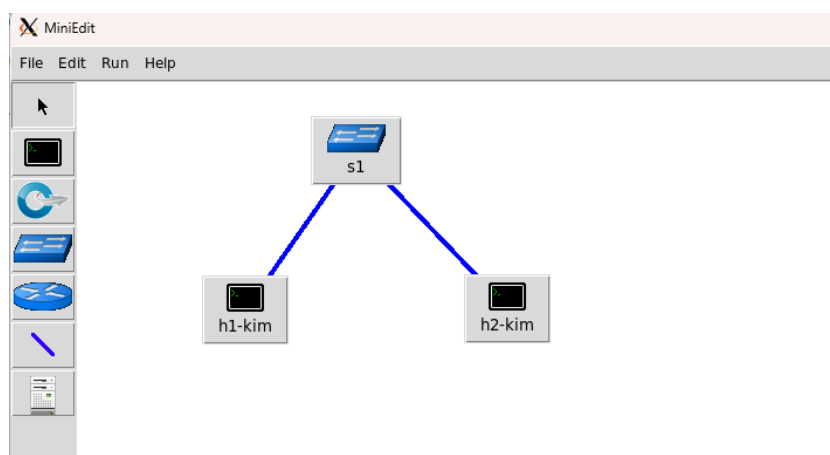


Рис. 2.18: Построение топологии

The image shows the 'Properties' dialog box for the host 'h1-kim' in the MiniEdit application. The dialog has four tabs: 'Properties', 'VLAN Interfaces', 'External Interfaces', and 'Private Directories'. The 'Properties' tab is currently selected. It contains several input fields: 'Hostname' with the value 'h1-kim', 'IP Address' with the value '10.0.0.1/8', 'Default Route' (empty), 'Amount CPU' (empty), 'Cores' (empty), 'Start Command' (empty), and 'Stop Command' (empty). To the right of the 'Amount CPU' field is a dropdown menu showing 'host'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Рис. 2.19: IP-адрес для хоста h1

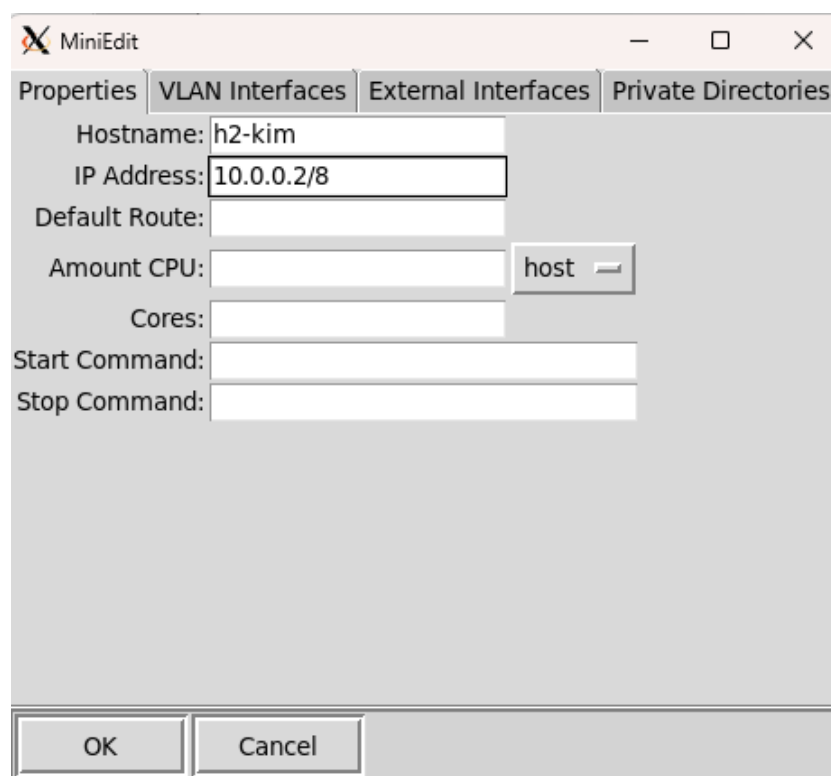


Рис. 2.20: IP-адрес для хоста h2

12. Проверка связности.



```
Host: h1-kim@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
h1-kim-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 c2:e4:23:0c:cd:30 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 853 bytes 228220 (228.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 853 bytes 228220 (228.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/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=0.327 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.047 ms
^C
--- 10.0.0.2 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13320ms
rtt min/avg/max/mdev = 0.030/0.062/0.327/0.074 ms
```

Рис. 2.21: Проверка связности ping

### 13. Автоматическое назначение IP-адресов

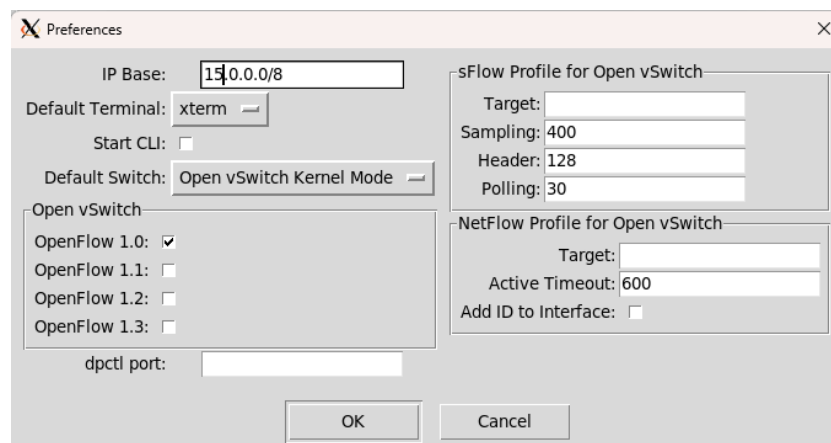


Рис. 2.22: Именование IP-адрес

```
Host: h1-kim" @mininet-vm
root@mininet-vm:/home/mininet# ifconfig
h1-kim-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 15.0.0.1 netmask 255.0.0.0 broadcast 15.255.255.255
    ether 9e:c0:8c:28:7b:b6 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 1141 bytes 247148 (247.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1141 bytes 247148 (247.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 2.23: Отображение IP-адресов

Интерфейс h1-eth0 на узле h1 теперь имеет IP-адрес 15.0.0.1 и маску подсети 255.0.0.0.

## 12. Сохранение и загрузка топологии Mininet.

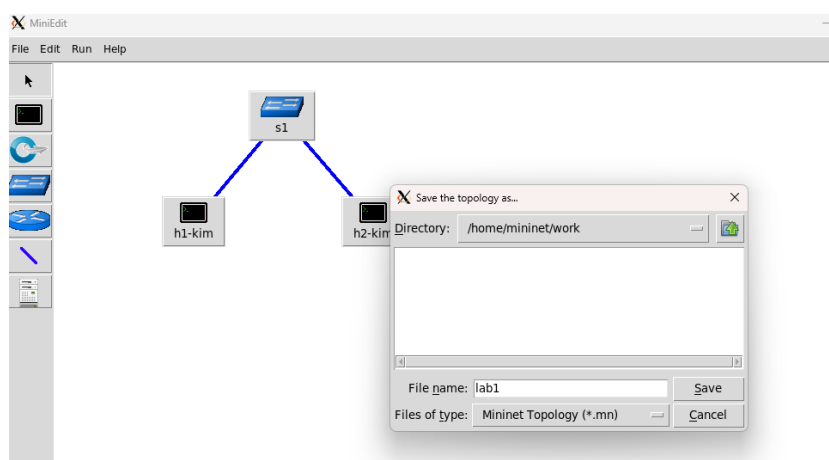


Рис. 2.24: Сохранение топологии

```
mininet@mininet-vm:~$ cd work/
mininet@mininet-vm:~/work$ ls -Al
total 4
-rw-r--r-- 1 root root 1671 Nov 16 02:36 lab1.mn
mininet@mininet-vm:~/work$ cd
mininet@mininet-vm:~$ sudo chown -R mininet:mininet ~/work
mininet@mininet-vm:~$
```

Рис. 2.25: Право доступа к файлу

## 3 Вывод

Я развернула в системе виртуализации mininet, знакомство с основными командами для работы с Mininet через командную строку и через графический интерфейс.