

Лабораторная работа №1

Введение в Mininet

Тазаева Анастасия Анатольевна

Содержание

1	Цель работы	6
2	Выполнение лабораторной работы	7
2.1	Настройка стенда виртуальной машины Mininet	7
2.1.1	Настройка образа VitruaBox	7
2.1.2	Подключение к виртуальной машине	9
2.1.3	Настройка доступа к Интернету	11
2.1.4	Обновление версии mininet	13
2.1.5	Настройка параметров XTerm	14
2.1.6	Настройка соединения X11 для суперпользователя	15
2.2	Основы работы в Mininet	16
2.2.1	Работа с Mininet с помощью командной строки	16
2.2.2	Построение и эмуляция сети в Mininet с использованием графического интерфейса	21
3	Выводы	27

Список иллюстраций

2.1	Виртуальная машина Mininet	8
2.2	Обнаруженные неправильные настройки	8
2.3	Исправление неправильных настроек. Изменение типа графического контроллера	8
2.4	Сеть 1. Подключение типа NAT	9
2.5	Сеть 2. Подключение host-only network adapter	9
2.6	Подключение к виртуальной машине. Команда ifconfig	10
2.7	Подключение к виртуальной машине через терминал хостовой машины	10
2.8	Подключение к виртуальной машине через терминал хостовой машины с настроенным ssh-подсоединением	11
2.9	Просмотр активных интерфейсов	11
2.10	Активация интерфейса NAT	12
2.11	Установка mc	12
2.12	Редактирование файла /etc/netplan/01-netcfg.yaml	13
2.13	Изменение названия предыдущей установки	13
2.14	Установка новой версии Mininet	13
2.15	Обновление исполняемых файлов	14
2.16	Проверка установленной версии mininet	14
2.17	Редактирование файла /etc/X11/app-defaults/XTerm	15
2.18	Попытка запуска приложения из-под суперпользователя. Ошибка	15
2.19	Значение куки (MIT magic cookie) 1 пользователя mininet	16
2.20	Заполнение файла полномочий скопированным значением куки	16
2.21	Значение куки (MIT magic cookie) пользователя root	16
2.22	Запуск приложения из-под суперпользователя	16
2.23	Команда sudo mn	17
2.24	Команда help	18
2.25	Команда nodes	18
2.26	Команда net	18
2.27	Конфигурация узла h1 и узла h2	19
2.28	Конфигурация узла s1	20
2.29	Проверка связности между узлами h1 и h2	20
2.30	Остановка эмуляции	21
2.31	MiniEdit	21
2.32	Топология	22
2.33	Свойства хоста h1	22
2.34	Свойства хоста h2	23

2.35 Run	24
2.36 h1. ifconfig. Проверка связности между узлами h1 и h2 в Mininet . .	24
2.37 h2. ifconfig	25
2.38 Edit Preferences	25
2.39 h1. ifconfig	26
2.40 h2. ifconfig	26
2.41 Редактирование каталога с топологиями	26

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 Настройка стенда виртуальной машины Mininet

2.1.1 Настройка образа VitrualBox

1. С репозитория Mininet был скачан и распакован актуальный релиз ovf-образа виртуальной машины.
2. Далее запустила VitrualBox и импортировала файл .ovf (рис. 2.1). Необходимо было проверить настройки и исправить неправильные, что я и сделала. В нашем случае нужно было изменить тип графического контроллера (рис. 2.2 и 2.3).

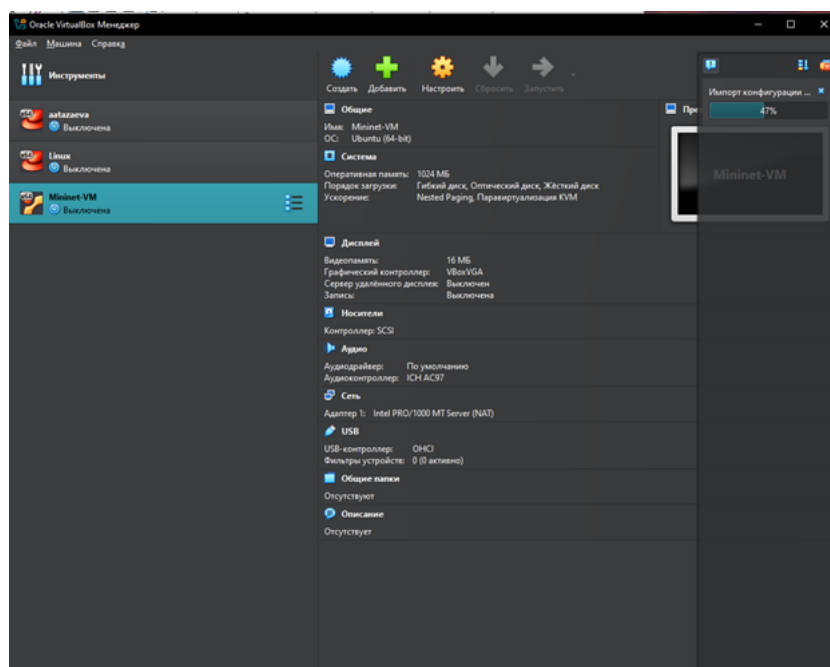


Рис. 2.1: Виртуальная машина Mininet

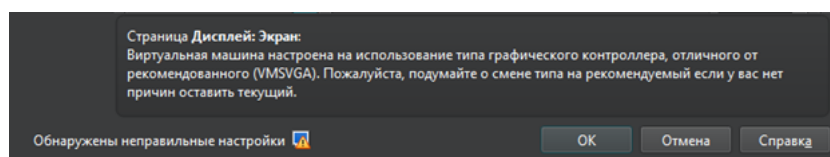


Рис. 2.2: Обнаруженные неправильные настройки

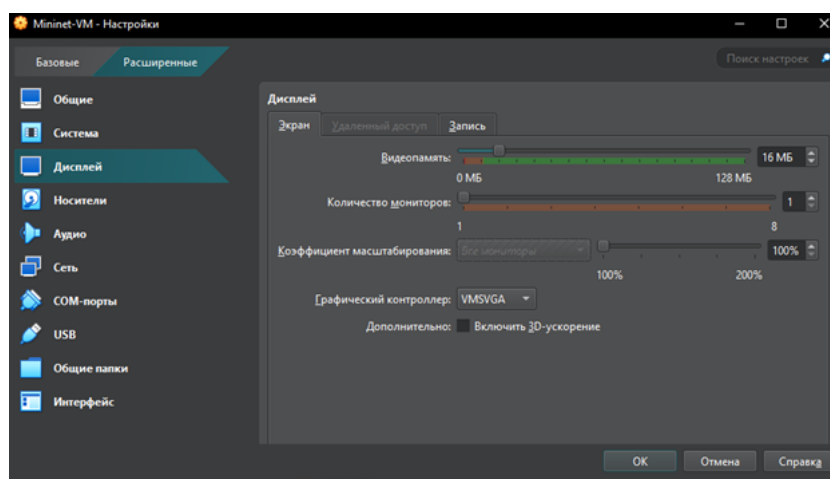


Рис. 2.3: Исправление неправильных настроек. Изменение типа графического контроллера

3. В настройках сети первый адаптер должен иметь подключение типа NAT (рис. 2.4). Для второго адаптера указала тип подключения host-only network adapter (виртуальный адаптер хоста) (рис. 2.4).

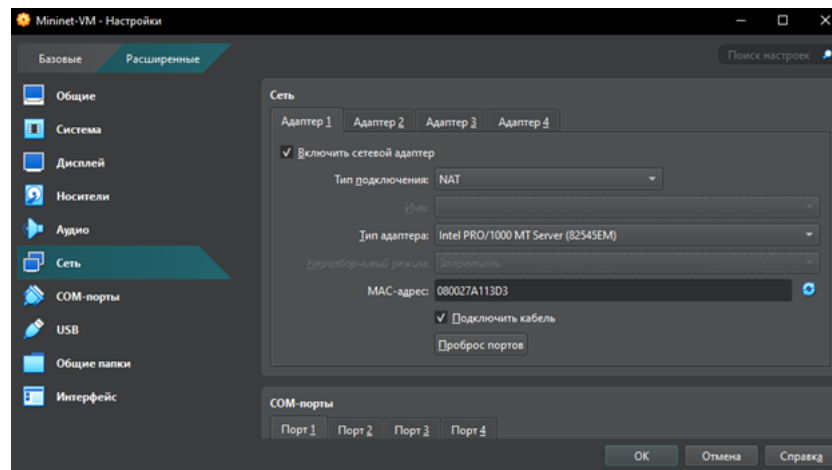


Рис. 2.4: Сеть 1. Подключение типа NAT

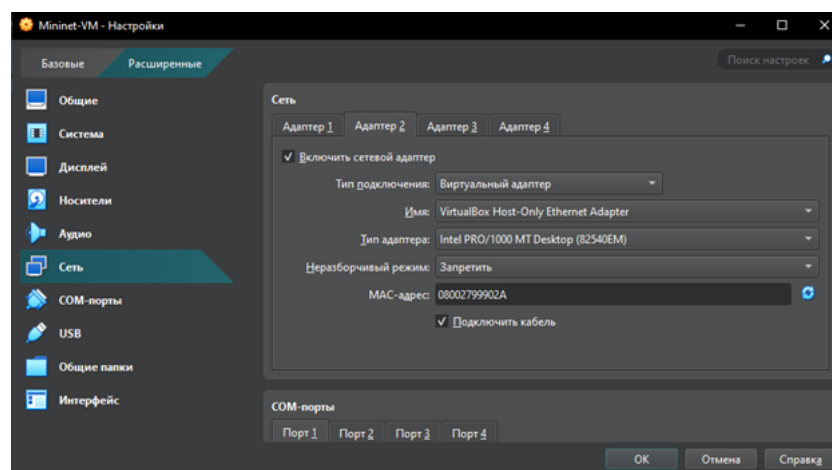


Рис. 2.5: Сеть 2. Подключение host-only network adapter

4. Запустила виртуальную машину с Mininet.

2.1.2 Подключение к виртуальной машине

1. Залогинилась в виртуальной машине, введя логин и пароль (рис. 2.6). Также просмотрела адрес машины - 192.168.56.101.

```

Ubuntu 20.04.1 LTS mininet-vm tty1
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 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:99:90:2a 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 48 bytes 3688 (3.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 48 bytes 3688 (3.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$ _

```

Рис. 2.6: Подключение к виртуальной машине. Команда ifconfig

2. Подключилась к виртуальной машине из терминала хостовой машины с помощью команды `ssh -Y mininet@192.168.56.101` (рис. 2.7). Отключилась с помощью комбинации клавиш `Ctrl + d`.

```

[aatazaeva@aatazaeva ~]$ ssh -Y mininet@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:mvosfu3mcoPZaycNRXq+i3xXMo7i5bTKEmv+rquiLCI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts
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

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

Last login: Sat Nov 16 04:23:56 2024
/usr/bin/xauth: file /home/mininet/.Xauthority does not exist
mininet@mininet-vm:~$ logout
Connection to 192.168.56.101 closed.

```

Рис. 2.7: Подключение к виртуальной машине через терминал хостовой машины

3. Настроила ssh-подсоединение по ключу к виртуальной машине, для чего в терминале основной Linux-машины перешла в каталог `.ssh` своего домашнего каталога, ввела `ssh-copy-id mininet@192.168.56.101`. Вновь

подключилась к виртуальной машине и убедилась, что подключение происходит успешно и без ввода пароля (рис. 2.8).

```
[aatazaeva@aatazaeva ~]$ ssh -Y mininet@192.168.56.101
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: Sat Nov 16 04:30:19 2024 from 192.168.56.1
mininet@mininet-vm:~$ S
```

Рис. 2.8: Подключение к виртуальной машине через терминал хостовой машины с настроенным ssh-подсоединением

2.1.3 Настройка доступа к Интернету

1. После подключения к виртуальной машине mininet посмотрела IP-адреса машины с помощью команды `ifconfig` (рис. 2.9). Для доступа к сети Интернет должен быть активен адрес NAT: 10.0.0.x. В нашем случае его нет, так что активировала интерфейс, набрав в командной строке `sudo dhclient eth1` (рис. 2.10).

```
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:99:90:2a txqueuelen 1000 (Ethernet)
    RX packets 151 bytes 31720 (31.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 27997 (27.9 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 432 bytes 33080 (33.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 432 bytes 33080 (33.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 2.9: Просмотр активных интерфейсов

```

mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:99:90:2a txqueuelen 1000 (Ethernet)
    RX packets 201 bytes 35508 (35.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 174 bytes 31359 (31.3 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:a1:13:d3 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 528 bytes 40456 (40.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 528 bytes 40456 (40.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 2.10: Активация интерфейса NAT

2. Для дальнейшей работы установила mc (рис. 2.11).

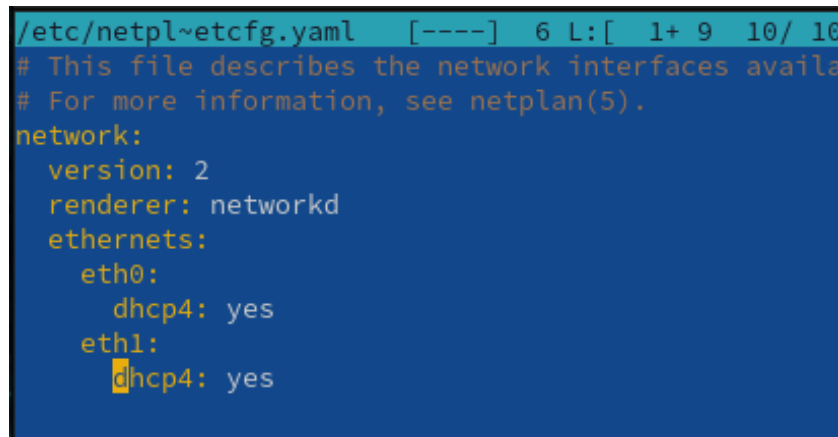
```

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
-2.1build1 [75.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 mc-data all 3:4.8.2
2ubuntu1 [1,265 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 mc amd64 3:4.8.24-2
ubuntu1 [477 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal/main amd64 unzip amd64 6.0-25ubunt
[169 kB]
Fetched 1,986 kB in 2s (1,030 kB/s)
Selecting previously unselected package libssh2-1:amd64.
(Reading database ... 101729 files and 1 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.
Preparing to unpack .../mc_3%3a4.8.24-2ubuntu1_amd64.deb ...
Unpacking mc (3:4.8.24-2ubuntu1) ...
Selecting previously unselected package unzip.
Preparing to unpack .../unzip_6.0-25ubuntu1_amd64.deb ...
Unpacking unzip (6.0-25ubuntu1) ...

```

Рис. 2.11: Установка mc

3. Для удобства дальнейшей работы добавила для mininet указание на использование двух адаптеров при запуске. Для этого требовалось перейти в режим суперпользователя и внести изменения в файл /etc/netplan/01-netcfg.yaml виртуальной машины mininet `sudo mcedit /etc/netplan/01-netcfg.yaml` (рис. 2.12).

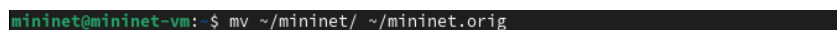


```
/etc/netpl~etcfg.yaml [----] 6 L:[ 1+ 9 10/ 10
# This file describes the network interfaces availa
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernet:
    eth0:
      dhcp4: yes
    eth1:
      dhcp4: yes
```

Рис. 2.12: Редактирование файла /etc/netplan/01-netcfg.yaml

2.1.4 Обновление версии mininet

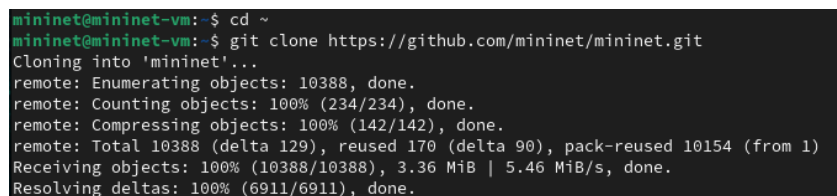
1. В виртуальной машине mininet переименовала предыдущую установку Mininet `mv ~/mininet ~/mininet.orig` (рис. 2.13).



```
mininet@mininet-vm: $ mv ~/mininet/ ~/mininet.orig
```

Рис. 2.13: Изменение названия предыдущей установки

2. Скачала новую версию Mininet (рис. 2.14).



```
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% (142/142), done.
remote: Total 10388 (delta 129), reused 170 (delta 90), pack-reused 10154 (from 1)
Receiving objects: 100% (10388/10388), 3.36 MiB | 5.46 MiB/s, done.
Resolving deltas: 100% (6911/6911), done.
```

Рис. 2.14: Установка новой версии Mininet

3. Обновила исполняемые файлы (рис. 2.15).

```
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 mi
ninet==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 s
ha256=8f26ca24ad074d6226de96eae71d29a9936ad2d0cf99216361e8ed266deec0bc
  Stored in directory: /tmp/pip-ephem-wheel-cache-lj2j8y05/wheels/cd/7d/a7/aafelb3ea
ff31efd6ba4e2ea6c9690a717bdf739db6cfe8d45
Successfully built mininet
Installing collected packages: mininet
Successfully installed mininet-2.3.1b4
```

Рис. 2.15: Обновление исполняемых файлов

4. Проверила номер установленной версии mininet (рис. 2.16). Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.1).

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

Рис. 2.16: Проверка установленной версии mininet

2.1.5 Настройка параметров XTerm

1. 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` (рис. 2.17). Выбран системный моноширинный шрифт, кегль шрифта — 12 пунктов.

```

/etc/X11/~lts/XTerm [----] 18 L:[228+39 267/267] *(10377/10377b) <EOF> [*][X]
! Eliminate the border of the buttons in the menubar, so the only line around
! the text is for the highlighted button:
*MenuBar*borderWidth: 0

! Set a border for the menus to make them simpler to distinguish against the
! vt100 widget:
*SimpleMenu*borderWidth: 2

! xterm can switch at runtime between bitmap (default) and TrueType fonts.
! The "faceSize" resource controls the size of the latter. However, it was
! originally given with a size that makes the two types of fonts different
! sizes. Uncomment this line to use the same size as "fixed".
!*faceSize: 8

! Here is a pattern that is useful for double-clicking on a URL:
*charClass: 33:48,35:48,37-38:48,43-47:48,58:48,61:48,63-64:48,95:48,126:48
!
! Alternatively,
!*on2Clicks: regex [[:alpha:]]+//[[:alnum:]]!#+,./=?@_~]|(%[[:xdigit:]][:xdigit:]]

! VT100s and similar terminals recognize escape sequences and control
! characters to which they reply to the host with other escape sequences,
! to provide information. The "resize" program uses this feature.
!
! In addition, xterm recognizes several escape sequences which can be used to
! set fonts, window properties, return settings via escape sequences. Some
! find these useful; others are concerned with the possibility of unexpected
! inputs.
!
! All of these features can be enabled or disabled via menus.
!
! 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

```

Рис. 2.17: Редактирование файла /etc/X11/app-defaults/XTerm

2.1.6 Настройка соединения X11 для суперпользователя

1. При попытке запуска приложения из-под суперпользователя возникает ошибка (рис. 2.18).

```

mininet@mininet-vm:~$ xterm
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xterm
X11 connection rejected because of wrong authentication.
Warning: This program is an suid-root program or is being run by the root user.
The full text of the error or warning message cannot be safely formatted
in this environment. You may get a more descriptive message by running the
program as a non-root user or by removing the suid bit on the executable.
xterm: Xt error: Can't open display: %s
root@mininet-vm:~#

```

Рис. 2.18: Попытка запуска приложения из-под суперпользователя. Ошибка

2. Для исправления этой ситуации заполнила файл полномочий /root/.Xauthority , используя утилиту xauth. Скопировала значение куки (MIT magic cookie) 1

пользователя mininet в файл для пользователя root (рис. 2.19, 2.20, 2.21, 2.22).

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  ac5c4991c2d28ac1b8128bae87bcf379
```

Рис. 2.19: Значение куки (MIT magic cookie) 1 пользователя mininet

```
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  ac5c4991c2d28ac1b8128bae87bcf379
```

Рис. 2.20: Заполнение файла полномочий скопированным значением куки

```
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  ac5c4991c2d28ac1b8128bae87bcf379
```

Рис. 2.21: Значение куки (MIT magic cookie) пользователя root



Рис. 2.22: Запуск приложения из-под суперпользователя

2.2 Основы работы в Mininet

2.2.1 Работа с Mininet с помощью командной строки

1. Для запуска минимальной топологии ввела в командной строке `sudo mn` (рис. 2.23). Эта команда запускает Mininet с минимальной топологией, состоящей

из коммутатора, подключённого к двум хостам.

```
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. 2.23: Команда sudo mn

2. Для отображения списка команд интерфейса командной строки Mininet и примеров их использования ввела команду в интерфейсе командной строки Mininet help(рис. 2.24).

```
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
```

Рис. 2.24: Команда help

3. Для отображения доступных узлов ввела nodes (рис. 2.25). Вывод этой команды показывает, что есть два хоста (хост h1 и хост h2) и коммутатор (s1).

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

Рис. 2.25: Команда nodes

4. Ввела команду net в интерфейсе командной строки Mininet(рис. 2.26), чтобы просмотреть доступные линки.

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

Рис. 2.26: Команда 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 .

5. Посмотрела конфигурацию всех узлов. h1 (рис. 2.27) - 10.0.0.1, h2 (рис. 2.27) - 10.0.0.2, s1 (рис. 2.28).

```
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 de:5a:99:1f:f6:14 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> 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 a6:06:fb:84:3f:75 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. 2.27: Конфигурация узла h1 и узла h2

```

mininet> s1 ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:99:90:2a txqueuelen 1000 (Ethernet)
    RX packets 2987 bytes 443157 (443.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2741 bytes 627763 (627.7 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:a1:13:d3 txqueuelen 1000 (Ethernet)
    RX packets 4597 bytes 5898209 (5.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1048 bytes 73720 (73.7 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 2458 bytes 607094 (607.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2458 bytes 607094 (607.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether ba:58:9e:91:cd:47 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

s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 42:af:f0:d5:8a:9a 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. 2.28: Конфигурация узла s1

6. Проверка связности. Чтобы проверить связь между узлами, использовала команду ping (рис. 2.29). Команда ping работает, отправляя сообщения эхо-запроса протокола управляющих сообщений Интернета (ICMP) на удалённый компьютер и ожидая ответа.

```

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=1.50 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.144 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.045 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 0.043/0.431/1.495/0.615 ms

```

Рис. 2.29: Проверка связности между узлами h1 и h2

7. Остановила эмуляцию (рис. 2.30).

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
. .
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 432.978 seconds
```

Рис. 2.30: Остановка эмуляции

2.2.2 Построение и эмуляция сети в Mininet с использованием графического интерфейса

1. В терминале виртуальной машины mininet запустила MiniEdit с помощью команды `sudo ~/mininet/mininet/examples/miniedit.py` (рис. 2.31).

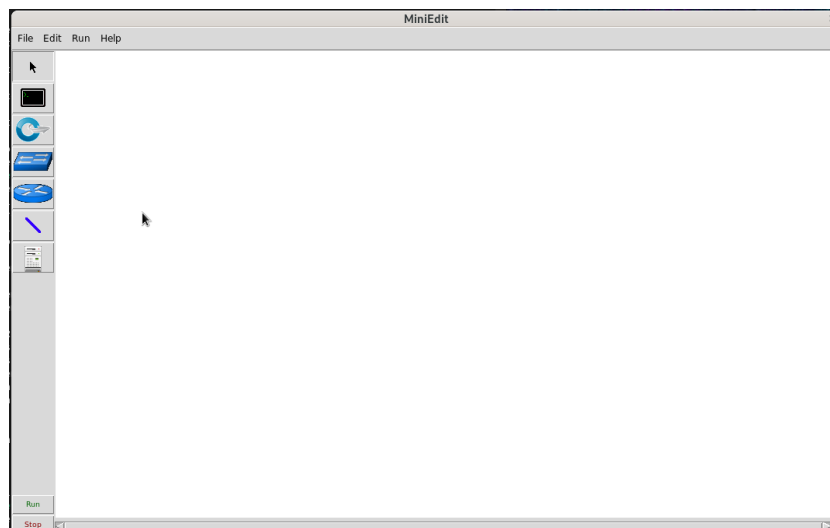


Рис. 2.31: MiniEdit

2. Добавила два хоста и один коммутатор, соединила хосты с коммутатором (рис. 2.32). Настроила IP-адреса на хостах h1 (рис. 2.33) и h2 (рис. 2.34).

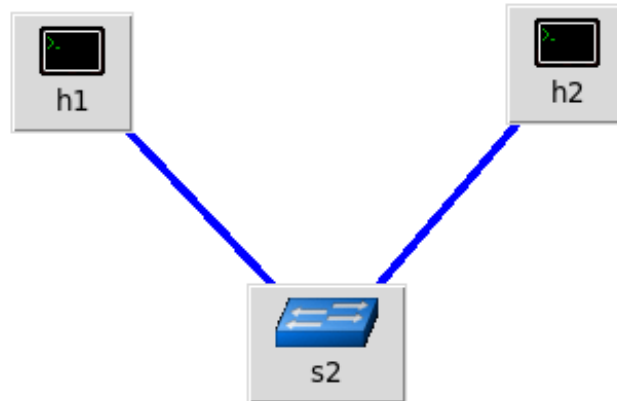


Рис. 2.32: Топология

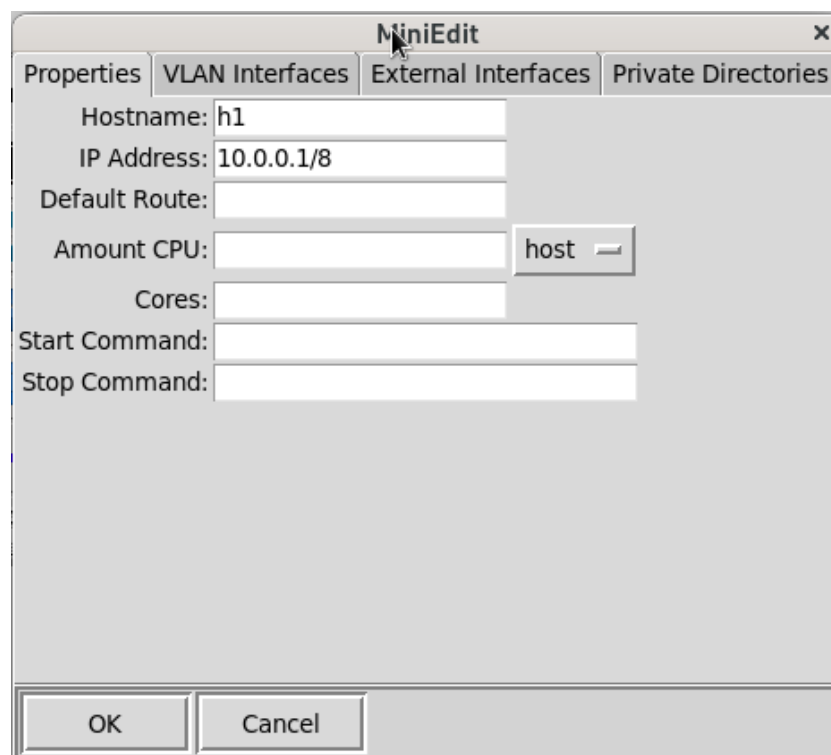


Рис. 2.33: Свойства хоста h1

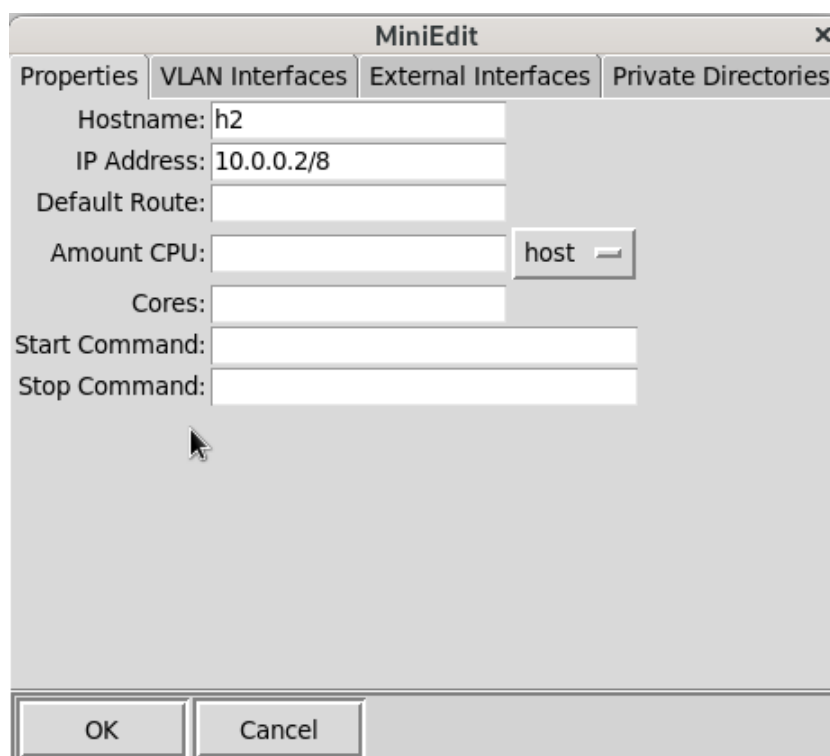


Рис. 2.34: Свойства хоста h2

3. Проверка связности. Запустила эмуляцию (рис. 2.35). И пропинговала с терминала хоста h1 хост h2 (рис. 2.36), предварительно просмотрев их адреса (рис. 2.36 и 2.37)

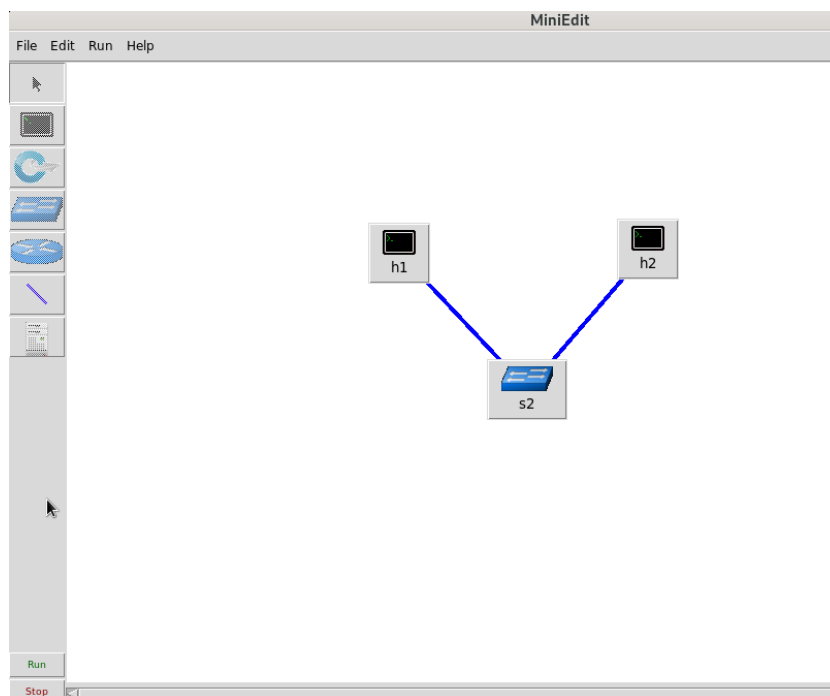


Рис. 2.35: Run

```

root@mininet-vm:/home/mininet# 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 6a:17:53:b2:7a:7e 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 914 bytes 258032 (258.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 914 bytes 258032 (258.0 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=0.172 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.045 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=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.042 ms

```

Рис. 2.36: h1. ifconfig. Проверка связности между узлами h1 и h2 в Mininet


```

root@mininet-vm:/home/mininet# 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 56:b7:2a:aa:b4:d0 txqueuelen 1000 (Ethernet)
    RX packets 7 bytes 574 (574.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7 bytes 574 (574.0 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 703 bytes 246516 (246.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 703 bytes 246516 (246.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 2.37: h2. ifconfig

3. 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. Удалила назначенные вручную адреса для хостов 1 и 2. В MiniEdit нажала Edit Preferences . По умолчанию в поле базовые значения IP-адресов (IP Base) установлено 10.0.0.0/8 . Изменила это значение на 15.0.0.0/8 (рис. 2.38). Запустила эмуляцию, отобразила IP-адреса, назначенные хосту h1 и h2 (рис. 2.39 и 2.40). Интерфейс h1-eth0 на узле h1 теперь имеет IP-адрес 15.0.0.1 и маску подсети 255.0.0.0.

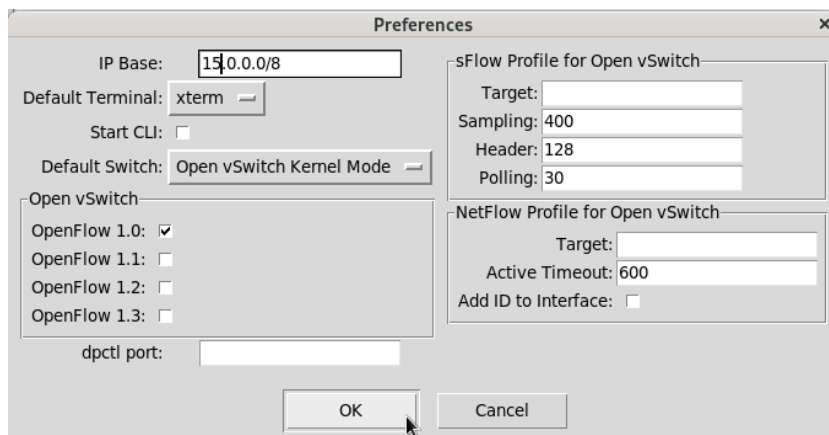


Рис. 2.38: Edit Preferences

```

"Host: h1" (на mininet-vm)
root@mininet-vm:/home/mininet# ifconfig
h1-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 96:b6:8b:dc:1c:d4 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 709 bytes 246700 (246.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 709 bytes 246700 (246.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 2.39: h1. ifconfig

```

"Host: h2" (на mininet-vm)
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 5a:04:91:f5:b5:ec 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 898 bytes 255228 (255.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 898 bytes 255228 (255.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 2.40: h2. ifconfig

4. Сохранение и загрузка топологии Mininet. Создала каталог work, сохранила топологию и поменяла права доступа к файлам в каталоге проекта (рис. 2.41).

```

mininet@mininet-vm:~/work$ ls -l
total 4
-rw-r--r-- 1 root root 1656 Nov 16 05:29 lab01.mn
mininet@mininet-vm:~/work$ cd
mininet@mininet-vm:~$ sudo chown -R mininet:mininet ~/work

```

Рис. 2.41: Редактирование каталога с топологиями

3 Выводы

В ходе лабораторной работы мною была развёрнута в системе виртуализации VirtualBox виртуальная среда mininet. Я познакомилась с основными командами для работы с Mininet через командную строку и через графический интерфейс.