

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина: Моделирование сетей передачи данных

Студент: Саинт-Амур Измаэль

Группа: НПИбд-01-20

МОСКВА

2023 г.

Постановка задачи

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.

Выполнение работы

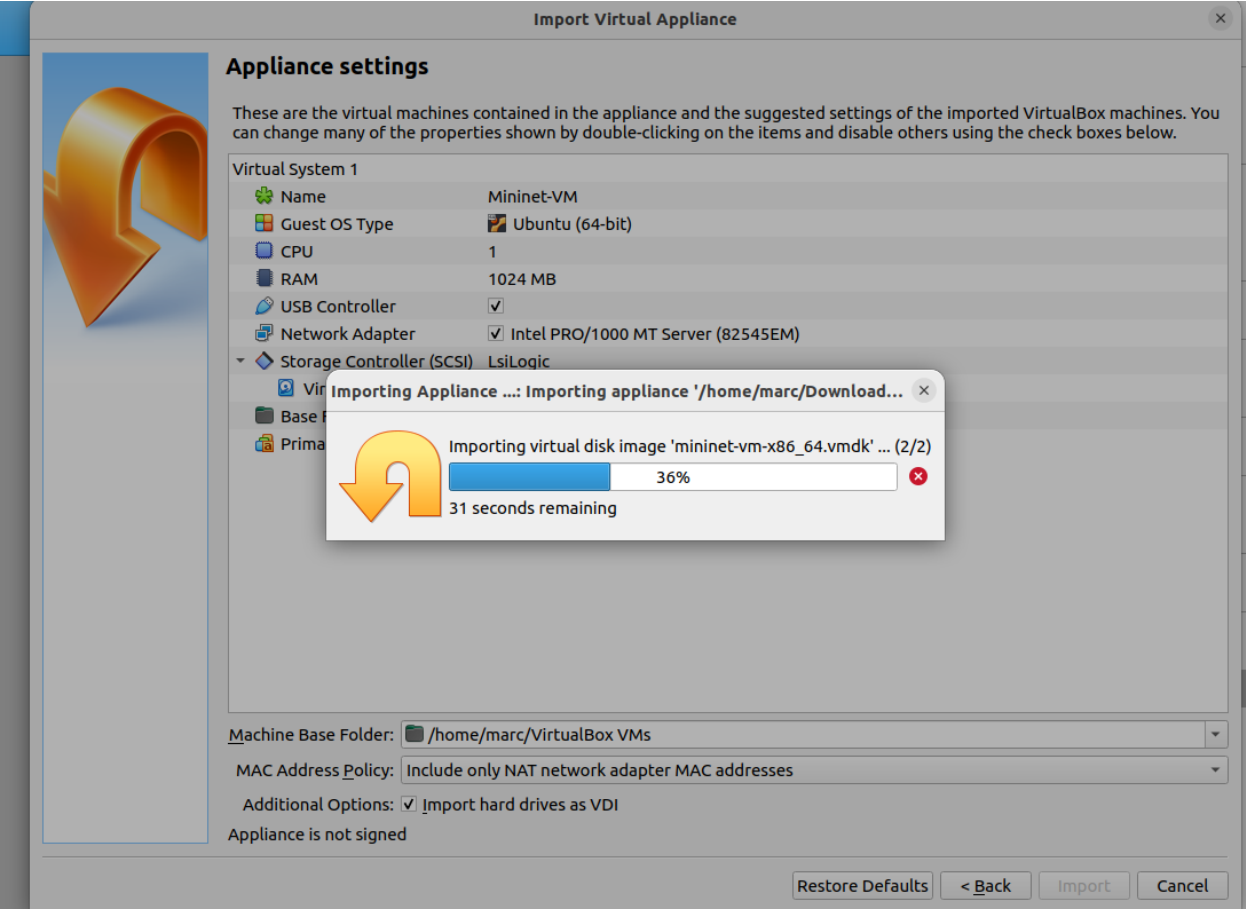
1.3.1. Настройка стенда виртуальной машины Mininet

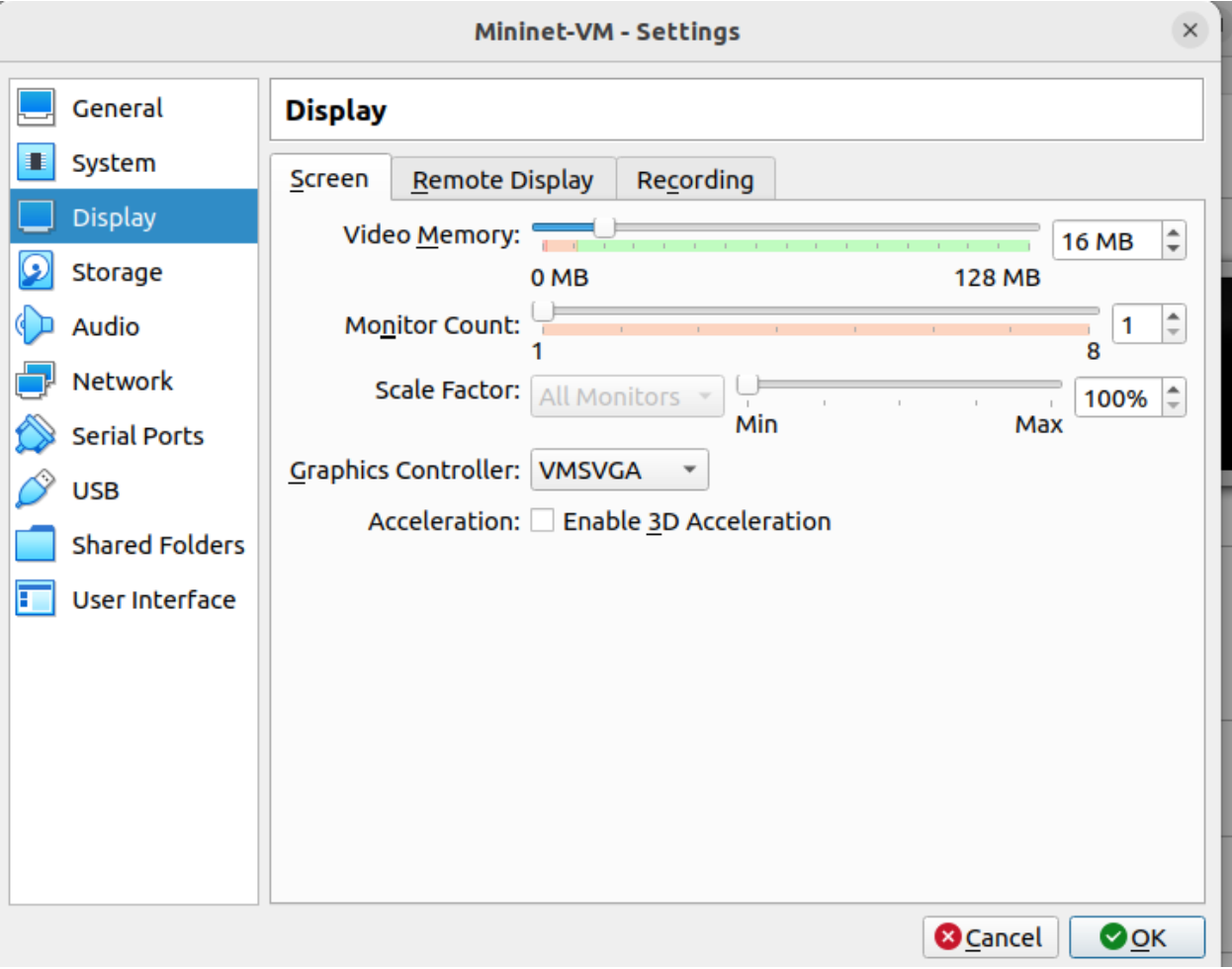
1.3.1.1. Общая информация

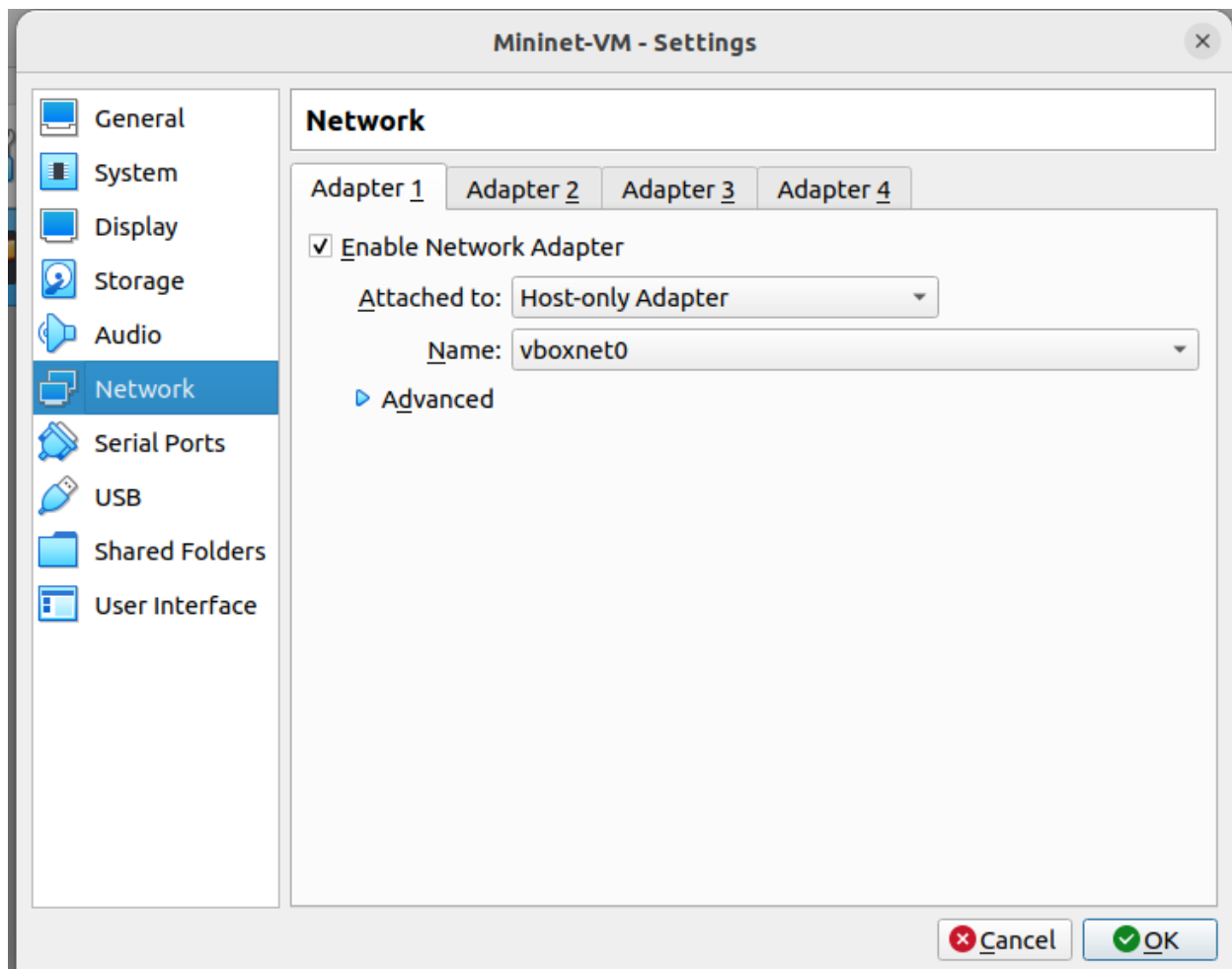
- Репозиторий Mininet: <https://github.com/mininet/mininet>
- Рекомендуемый к установке образ виртуальной машины: mininet-2.3.0-210211-ubuntu-20.04.1-legacy-server-amd64-ovf

1.3.1.2. Настройка образа VirtualBox

- Перейдите в репозиторий Mininet. Скачайте актуальный релиз ovf-образа виртуальной машины.
 - При необходимости переместите скачанный образ в каталог для работы, затем распакуйте его.
- Запустите систему виртуализации и импортируйте файл .ovf.
 - Перейдите в настройки системы виртуализации и уточните параметры настройки виртуальной машины. В частности, для VirtualBox выберете импортированную виртуальную машину и перейдите в меню Машина Настроить .







1.3.1.3. Подключение к виртуальной машине

– Залогиньтесь в виртуальной машине:

– login: mininet

– password: mininet

– Посмотрите адрес машины:

ifconfig

```

Login incorrect
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 08:00:27:89:81:b0  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 80  bytes 6128 (6.1 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 80  bytes 6128 (6.1 KB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$

```

- Внутренний адрес машины будет иметь вид 192.168.x.y.
 - Подключитесь к виртуальной машине (из терминала хостовой машины):
- 1 ssh -Y mininet@192.168.56.101

```

marc@ismael:~$ 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:kYtl1pkhiFa7KDXq7psRCT3RyTFeubuA3Y05d68duQ0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts
.
mininet@192.168.56.101's password:
Permission denied, please try again.
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: Fri Dec  1 20:15:27 2023
/usr/bin/xauth:  file /home/mininet/.Xauthority does not exist
mininet@mininet-vm:~$

```

- Для отключения ssh-соединения с виртуальной машиной нажмите Ctrl + d .
- Настройте ssh-подсоединение по ключу к виртуальной машине, для чего

в терминале основной Linux-машины перейдите в каталог `.ssh` своего домашнего каталога и введите (вместо `192.168.x.y` укажите внутренний адрес

виртуальной машины Mininet):

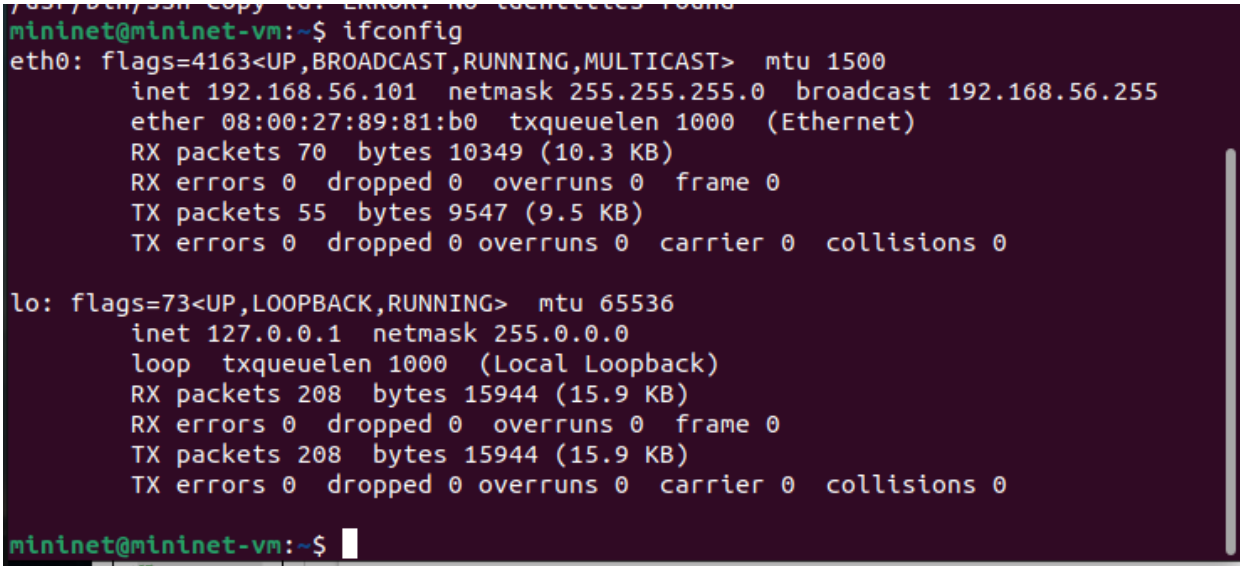
```
1 ssh-copy-id mininet@192.168.x.y
```

– Вновь подключитесь к виртуальной машине и убедитесь, что подключение происходит успешно и без ввода пароля.

1.3.1.4. Настройка доступа к Интернет

– После подключения к виртуальной машине mininet посмотрите IP-адреса машины:

```
ifconfig
```



```
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 08:00:27:89:81:b0  txqueuelen 1000  (Ethernet)
    RX packets 70  bytes 10349 (10.3 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 55  bytes 9547 (9.5 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 208  bytes 15944 (15.9 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 208  bytes 15944 (15.9 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$
```

– 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:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.15 netmask 255.255.255.0 broadcast 10.0.3.255
    ether 08:00:27:1c:e1:74 txqueuelen 1000 (Ethernet)
    RX packets 201 bytes 19654 (19.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 190 bytes 17432 (17.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:89:81:b0 txqueuelen 1000 (Ethernet)
    RX packets 90 bytes 6810 (6.8 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 24 bytes 2424 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 2424 (2.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$

```

– Для удобства дальнейшей работы установите mc:

```
1 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 l texlive-binaries dbview djvulibre-bin epub-utils genisoimage gv imagemagick
  libaspell-dev links l w3m l lynx odt2txt poppler-utils python python-boto python-tz xpdf
  l 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] _

```

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:

```
1 sudo mcedit /etc/netplan/01-netcfg.yaml
```

– В результате файл /etc/netplan/01-netcfg.yaml должен иметь следующий вид:


```
Mininet-VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
etc/netplan/01-netcfg.yaml [-M--] 15 L: 1+ 9 10/ 101 *(218 / 218b) <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
```

1.3.1.5. Обновление версии Mininet

– В виртуальной машине mininet переименуйте предыдущую установку Mininet:

```
1 mv ~/mininet ~/mininet.orig
```

– Скачайте новую версию Mininet:

```
cd ~
```

```
2git clone https://github.com/mininet/mininet.git
```

```
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% (234/234), done.
remote: Compressing objects: 100% (139/139), done.
remote: Total 10388 (delta 129), reused 176 (delta 93), pack-reused 10154
Receiving objects: 100% (10388/10388), 3.36 MiB | 2.33 MiB/s, done.
Resolving deltas: 100% (6911/6911), done.
mininet@mininet-vm:~$ cd ~/mininet
mininet@mininet-vm:~/mininet$ sudo make install
cc -Wall -Wextra \
-DVERSION=\\"PYTHONPATH=. python -B bin/mn --version Z>&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
-
```

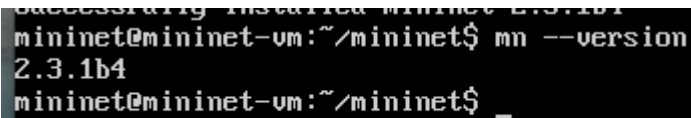
– Обновите исполняемые файлы:

```
cd ~/mininet
```

```
sudo make install
```

– Проверьте номер установленной версии mininet:

```
mn --version
```



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

1.3.1.6. 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. Для

этого можно воспользоваться следующей командой:

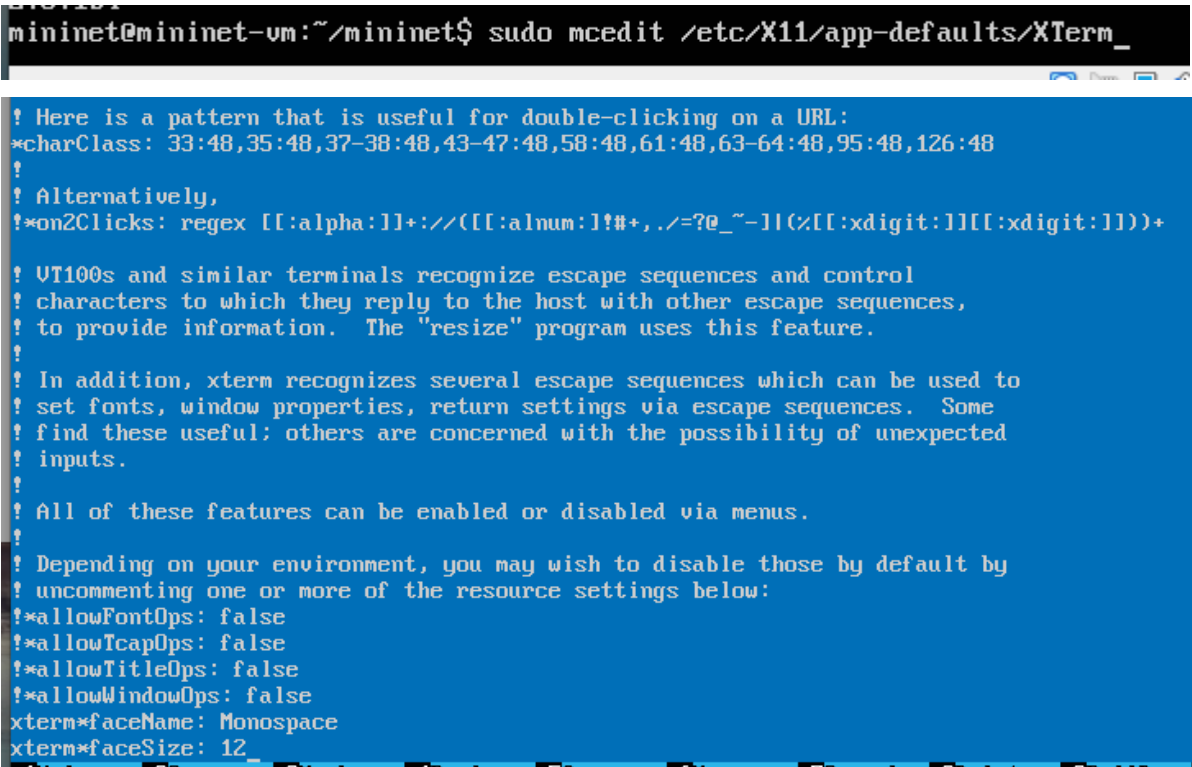
```
sudo mcedit /etc/X11/app-defaults/XTerm
```

и затем в конце файла добавить строки:

```
1 xterm*faceName: Monospace
```

```
2 xterm*faceSize: 12
```

Здесь выбран системный моноширинный шрифт, кегль шрифта — 12 пунктов.



```
mininet@mininet-vm:~/mininet$ sudo mcedit /etc/X11/app-defaults/XTerm_
! 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
```

1.3.1.7. Настройка соединения X11 для суперпользователя

При попытке запуска приложения из-под суперпользователя возникает ошибка:

X11 connection rejected because of wrong authentication.

Ошибка возникает из-за того, что X-соединение выполняется от имени пользователя mininet, а приложение запускается от имени пользователя root

с использованием sudo. Для исправления этой ситуации необходимо заполнить файл полномочий /root/.Xauthority, используя утилиту xauth.

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

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 c4e54a76c562effe7a11b708ecde1d17
mininet@mininet-vm:~$ 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 c4e54a76c562effe7a11b708ecde1d17
xauth: file /root/.Xauthority does not exist
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 c4e54a76c562effe7a11b708ecde1d17
root@mininet-vm:~# logout
mininet@mininet-vm:~$
```

1.3.2. Основы работы в Mininet

1.3.2.1. Работа с Mininet с помощью командной строки

. Вызов Mininet с использованием топологии по умолчанию.

– Для запуска минимальной топологии введите в командной строке:

```
sudo mn
```

Эта команда запускает Mininet с минимальной топологией, состоящей из коммутатора, подключённого к двум хостам.

– 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
```

Вывод этой команды показывает, что есть два хоста (хост h1 и хост h2) и коммутатор (s1).

– Иногда бывает полезно отобразить связи между устройствами в Mininet, чтобы понять топологию. 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>

```

Adapter 2: Intel PRO/1000 MT Desktop (N

```

=====
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>
```

```

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
mininet>

```

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.

- Посмотрите конфигурацию всех узлов. В отчёте укажите, какие адреса присвоены интерфейсам устройств.

2. 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

```
h1 ping 10.0.0.2
```

проверяет соединение между хостами h1 и h2. Для остановки теста нажмите

Ctrl + c .

. Остановка эмуляции.

Для завершения работы режима эмуляции mininet используйте команду

```
exit
```

Заметим, что команда sudo mn -c часто используется в терминале для очистки предыдущего экземпляра Mininet (например, после сбоя).

```

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 12:3e:01:55:ed: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> 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=27.6 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.619 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.143 ms

```

```

^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7127ms
rtt min/avg/max/mdev = 0.104/3.625/27.631/9.074 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 214.202 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 udpbwtest mnexec ivs ryu-manager 2> /dev/nul
l
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openfl
owd ovs-controller ovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/

```

1.3.2.2. Построение и эмуляция сети в Mininet с использованием графического интерфейса

1. Построение топологии сети.

– В терминале виртуальной машины mininet запустите MiniEdit:

`sudo ~/mininet/mininet/examples/miniedit.py`

```

mininet@mininet-vm:~$ ls -F ~/mininet/
CONTRIBUTORS  Makefile  custom/  examples/  mnexec*  setup.py
INSTALL        README.md  debian/  mininet/  mnexec.1  util/
LICENSE        bin/      doc/     mn.1      mnexec.c

mininet@mininet-vm:~$ ls -F ~/mininet/mininet/
__init__.py  cli.py  log.py  node.py  test/  util.py
__main__.py@  examples@  moduledeps.py  nodelib.py  topo.py
clean.py*  link.py  net.py*  term.py  toplib.py

mininet@mininet-vm:~$ ls -F ~/mininet/mininet/examples/
README.md  controllers.py*  miniedit.py*  popenpoll.py*
__init__.py  controllers2.py*  mobility.py*  scratchnet.py*
baresshd.py*  controlnet.py*  multilink.py*  scratchnetuser.py*
bind.py*  cpu.py*  multiping.py*  simpleperf.py*
cluster.py*  emptynet.py*  multipoll.py*  sshd.py*
clusterSanity.py*  hwintf.py*  multitest.py*  test/
clustercli.py  intfoptions.py*  nat.py*  tree1024.py*
clusterdemo.py*  limit.py*  natnet.py*  treeping64.py*
clusterperf.py*  linearbandwidth.py*  numberedports.py*  vlanhost.py*
consoles.py*  linuxrouter.py*  popen.py*

mininet@mininet-vm:~$ sudo ~/mininet/mininet/examples/miniedit.py

```

– Добавьте два хоста и один коммутатор, соедините хосты с коммутатором.

– Настройте IP-адреса на хостах h1 и h2. Для этого удерживая правую кнопку мыши на устройстве выберите свойства. Для хоста h1 укажите IP-адрес 10.0.0.1/8, а для хоста h2 — 10.0.0.2/8.

2. Проверка связности.

– Перед проверкой соединения между хостом h1 и хостом h2 необходимо запустить эмуляцию. Для запуска эмуляции нажмите кнопку Run. После начала эмуляции кнопки панели MiniEdit станут серыми, указывая на то, что в настоящее время они отключены.

– Откройте терминал на хосте h1, удерживая правую кнопку мыши на хосте h1 и выбрав Terminal. Это действие позволит выполнять команды на хосте h1.

– Откройте терминал на хосте h2.

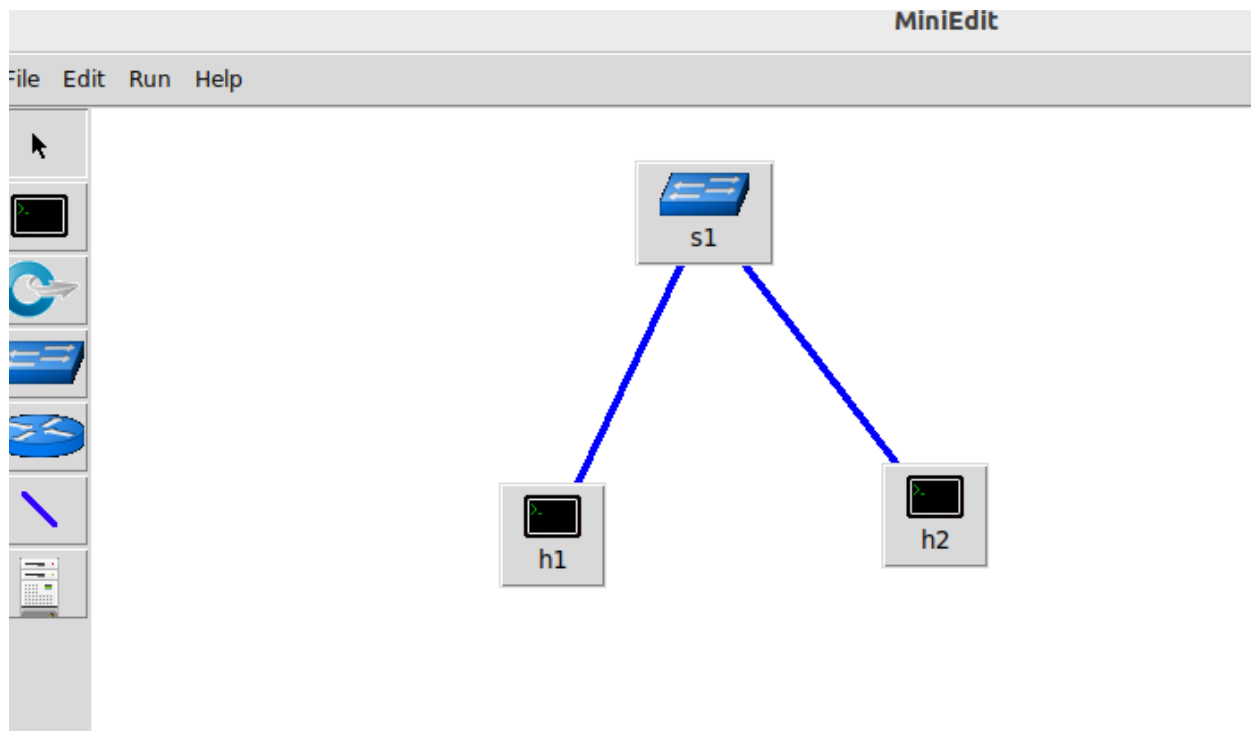
– На терминале хоста h1 введите команду `ifconfig`, чтобы отобразить на ☐значенные ему IP-адреса. Интерфейс h1-eth0 на хосте h1 должен быть

настроен с IP-адресом 10.0.0.1 и маской подсети 255.0.0.0.

– Повторите эти действия на хосте h2. Его интерфейс h2-eth0 должен быть настроен с IP-адресом 10.0.0.2 и маской подсети 255.0.0.0.

– Проверьте соединение между хостами, введя в терминале хоста h1 команду `ping 10.0.0.2`. Для остановки теста нажмите `Ctrl + c`.

– Остановите эмуляцию, нажав кнопку Stop.



MiniEdit

Properties | VLAN Interfaces | External Interfaces | Private Directories

Hostname:

IP Address:

Default Route:

Amount CPU: host

Cores:

Start Command:

Stop Command:


```

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 e2:b2:70:c3:52:71 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 999 bytes 3687008 (3.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 999 bytes 3687008 (3.6 MB)
    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=0.527 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.120 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.240 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.108 ms
^C

```

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.

- 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.

- Запустите эмуляцию, нажав кнопку Run.

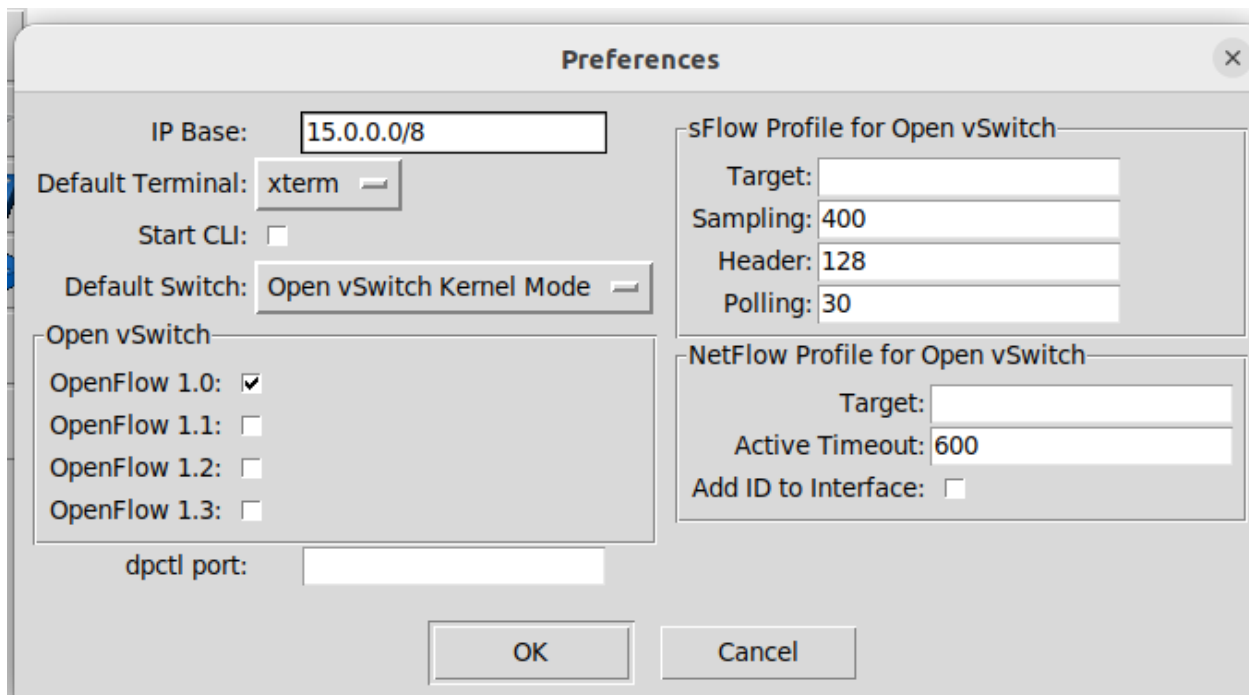
- Откройте терминал на хосте h1, удерживая правую кнопку мыши на хосте h1 и выбрав Terminal.

- Чтобы отобразить IP-адреса, назначенные хосту h1, введите команду
h1 ifconfig

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

- Вы также можете проверить IP-адрес, назначенный хосту h2. Соответствующий интерфейс h2-eth0 на хосте h2 должен иметь IP-адрес 15.0.0.2 и маску подсети 255.0.0.0.

- Остановите эмуляцию, нажав кнопку Stop



```
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 ba:c2:0e:5e:bb:43 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 952 bytes 3684564 (3.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 952 bytes 3684564 (3.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Сохранение и загрузка топологии Mininet.

– В домашнем каталоге виртуальной машины mininet создайте каталог для работы с проектами mininet:

```
1 mkdir ~/work
```

– Для сохранения топологии сети в файл нажмите в MiniEdit File Save .

Укажите имя для топологии и сохраните на своём компьютере.

– После сохранения проекта поменяйте права доступа к файлам в каталоге проекта:

1 sudo chown -R mininet:mininet ~/work

– Для загрузки топологии в MiniEdit нажмите File Open .

5. Завершите соединение с виртуальной машиной mininet и выключите её.

```
mininet@mininet-vm:~$ mkdir ~/work
mininet@mininet-vm:~$ sudo chown -R mininet:mininet ~/work
mininet@mininet-vm:~$
```

Выводы

В ходе полученные навыки развёртывание в системе виртуализации (например, в VirtualBox) mininet, знакомство с основными командами для работы с Mininet через командную строку и через графический интерфейс.