

Отчёт по лабораторной работе №1

Моделирование сетей передачи данных

Ищенко Ирина Олеговна

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Выводы	21

Список иллюстраций

2.1 Создание ВМ	6
2.2 Настройка сети	7
2.3 Запуск mininet	8
2.4 Подключение к ВМ с mininet через SSH	8
2.5 Просмотр IP-адресов машины	9
2.6 Активация второго интерфейса	9
2.7 Файл /etc/netplan/01-netcfg.yaml	10
2.8 Обновление Mininet	11
2.9 Настройка шрифтов XTerm	11
2.10 Опция перенаправления X11	12
2.11 Настройка соединения X11 для суперпользователя	12
2.12 Работа с Mininet с помощью командной строки	14
2.13 Проверка связности хостов	16
2.14 Проверка IP-адресов. Пинг	17
2.15 Смена базового IP-адреса	18
2.16 Просмотр IP-адреса на h1	19
2.17 Сохранение топологии	19

Список таблиц

1 Цель работы

Основной целью работы является развёртывание в системе виртуализации VirtualBox mininet, знакомство с основными командами для работы с Mininet через командную строку и через графический интерфейс.

2 Выполнение лабораторной работы

Перейдем в репозиторий Mininet, скачаем актуальный релиз ovf-образа виртуальной машины. Запустим систему виртуализации и импортируем файл .ovf и укажем параметры импорта (рис. 2.1).

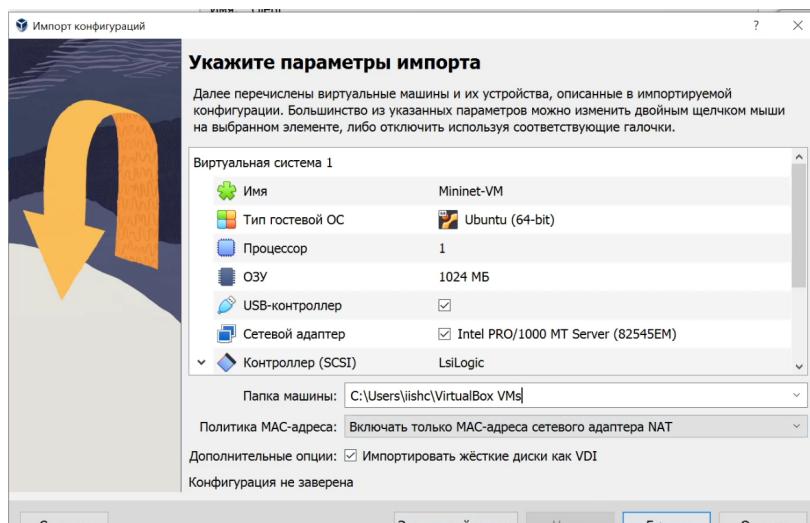


Рисунок 2.1: Создание ВМ

Перейдем в настройки системы виртуализации и уточним параметры настройки виртуальной машины. В частности, для VirtualBox выберем импортированную виртуальную машину и перейдите в меню «Машина -> Настроить».

Перейдем к опции «Система». Если внизу этого окна есть сообщение об обнаружении неправильных настроек, то, следуя рекомендациям, внесем исправления (изменим тип графического контроллера на рекомендуемый).

В настройках сети первый адаптер должен иметь подключение типа NAT, настроено по умолчанию. Для второго адаптера укажем тип подключения host-only network adapter (виртуальный адаптер хоста), который в дальнейшем мы будем использовать для входа в образ виртуальной машины (рис. 2.2).

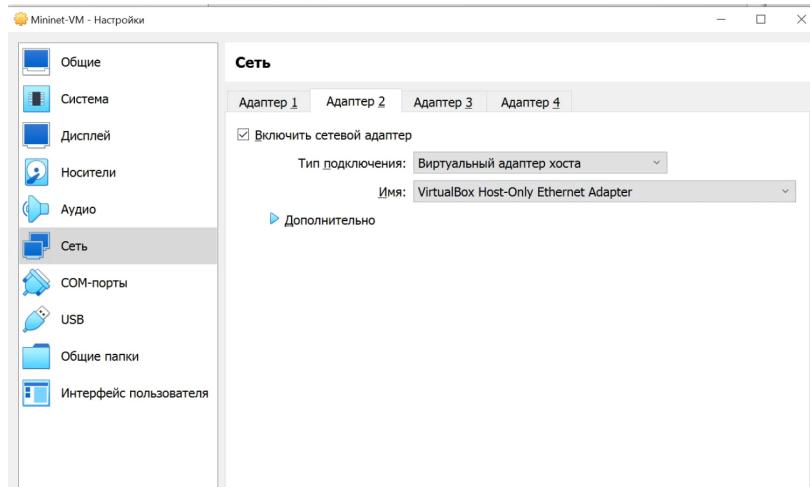


Рисунок 2.2: Настройка сети

Запустим виртуальную машину с Mininet. Залогинимся в виртуальную машину: - login: mininet - password: mininet

Посмотрим адрес машины с помощью ifconfig (рис. 2.3).

```

Ubuntu 20.04.1 LTS mininet-vm tty4

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

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

Last login: Tue Sep  9 10:39:24 PDT 2025 on tty1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
              ether 08:00:27:06:0f:9f txqueuelen 1000 (Ethernet)
                RX packets 3 bytes 1240 (1.2 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 88 bytes 6752 (6.7 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 88 bytes 6752 (6.7 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$
```

Рисунок 2.3: Запуск mininet

Подключимся к виртуальной машине (из терминала хостовой машины) (рис. 2.4).

```

C:\Users\iishc>ssh -Y mininet@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:6Q72Or1sYANBK95Ew4LQZIUzgvbI3G9iZ8Zl8vxsNiM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known hosts.
[mininet@192.168.56.102'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: Tue Sep  9 10:40:09 2025
mininet@mininet-vm:~$
```

Рисунок 2.4: Подключение к ВМ с mininet через SSH

После подключения к виртуальной машине mininet посмотрим IP-адреса машины (рис. 2.5). Активен только внутренний адрес машины 192.168.56.102, поэтому активируем второй интерфейс (рис. 2.6). Установим mc для удобства работы.

```

C:\Users\iishc>ssh -Y mininet@192.168.56.102
mininet@192.168.56.102's password:
Permission denied, please try again.
mininet@192.168.56.102's password:
Permission denied, please try again.
mininet@192.168.56.102'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
tion or proxy settings

Last login: Tue Sep  9 10:42:47 2025 from 192.168.56.1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
                ether 08:00:27:06:0f:9f  txqueuelen 1000  (Ethernet)
                RX packets 74  bytes 13481 (13.4 KB)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 70  bytes 12784 (12.7 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 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

mininet@mininet-vm:~$
```

Рисунок 2.5: Просмотр IP-адресов машины

```

mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
                ether 08:00:27:06:0f:9f  txqueuelen 1000  (Ethernet)
                RX packets 137  bytes 18223 (18.2 KB)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 103  bytes 16762 (16.7 KB)
                TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

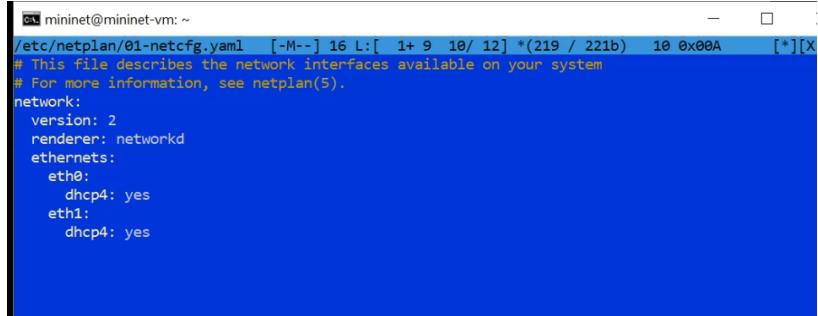
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
                ether 08:00:27:76:68:42  txqueuelen 1000  (Ethernet)
                RX packets 3  bytes 1770 (1.7 KB)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 3  bytes 1026 (1.0 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 568  bytes 43520 (43.5 KB)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 568  bytes 43520 (43.5 KB)
                TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

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 djvullibre-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] v
```

Рисунок 2.6: Активация второго интерфейса

Для удобства дальнейшей работы добавим для mininet указание на использование двух адаптеров при запуске. Для этого требуется перейти в режим суперпользователя и внести изменения в файл /etc/netplan/01-netcfg.yaml виртуальной машины mininet. В результате файл /etc/netplan/01-netcfg.yaml должен иметь следующий вид (рис. 2.7).



```
mininet@mininet-vm: ~
/etc/netplan/01-netcfg.yaml  [-M--] 16 L:[ 1+ 9 10/ 12] *(219 / 221b)  10 0x00A [*][X]
# 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
```

Рисунок 2.7: Файл /etc/netplan/01-netcfg.yaml

В виртуальной машине mininet переименуем предыдущую установку Mininet. Скачаем новую версию Mininet. Обновим исполняемые файлы. Проверим номер установленной версии mininet (рис. 2.8).

```

mininet@mininet-vm:~$ mv ~/mininet ~/mininet.orig
mininet@mininet-vm:~$ cd ~
mininet@mininet-vm:~$ git clone https://github.com/mininet/mininet.git
Cloning into 'mininet'...
remote: Enumerating objects: 10388, done.
remote: Counting objects: 100% (128/128), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 10388 (delta 102), reused 69 (delta 69), pack-reused 10260 (from 3)
Receiving objects: 100% (10388/10388), 3.36 MiB | 219.00 KiB/s, done.
Resolving deltas: 100% (6906/6906), 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==2.3.0b4) (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=71e9132b8f58d6a855426587b9c8f556b3cb998719902048aad25fd63aa2603a
  Stored in directory: /tmp/pip-ephem-wheel-cache-sdxvksef/wheels/cd/7d/a7/aafe1b3eaff31efd6ba4e2ea6c9690a717bdf739db6fe8d45
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$ 

```

Рисунок 2.8: Обновление Mininet

Настроим параметры XTerm для увеличения размера шрифта и применения векторных шрифтов вместо растровых. Внесем изменения в файл /etc/X11/app-defaults/XTerm (рис. 2.9).

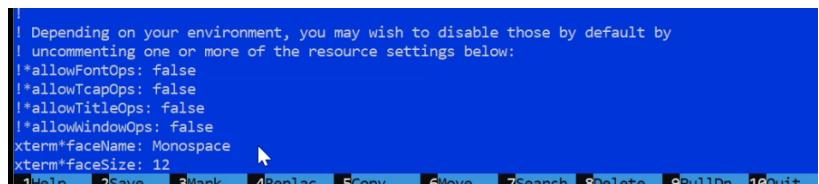


Рисунок 2.9: Настройка шрифтов XTerm

Установим putty и VcXsrv Windows X Server. Запустим XLaunch.

Выберем опции: - Multiple windows; - Display number: -1; - Start no client.

При подключении добавим опцию перенаправления X11: – Connection SSH X11 : Enable X11 forwarding (рис. 2.10).

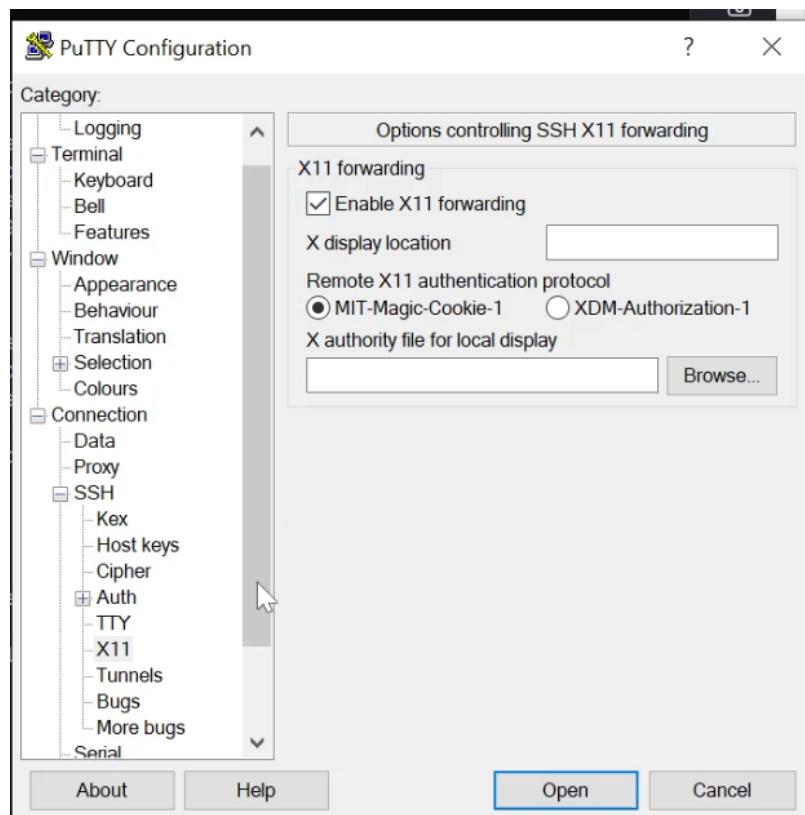


Рисунок 2.10: Опция перенаправления X11

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`. Скопируем значение куки (MIT magic cookie)1 пользователя `mininet` в файл для пользователя `root` (рис. 2.11).

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 f011e2d40ce0blef190090ecd3327810
mininet@mininet-vm:~$ ^C
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 f011e2d40ce0blef190090ecd3327810
xauth: file /root/.Xauthority does not exist
root@mininet-vm:~# xauth list
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 f011e2d40ce0blef190090ecd3327810
root@mininet-vm:~# logout
```

Рисунок 2.11: Настройка соединения X11 для суперпользователя

Для запуска минимальной топологии введем 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 введем команду в интерфейсе командной строки Mininet: help.

Для отображения доступных узлов введем: nodes (рис. 2.12). Вывод этой команды показывает, что есть два хоста (хост h1 и хост h2) и коммутатор (s1).



```
mininet@mininet-vm: ~
mininet@192.168.56.102'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: Tue Sep  9 10:52:28 2025 from 192.168.56.1
/usr/bin/xauth: file /home/mininet/.Xauthority does not exist
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
mininet>
```

Рисунок 2.12: Работа с Mininet с помощью командной строки

Иногда бывает полезно отобразить связи между устройствами в Mininet, чтобы понять топологию. Введем команду `net` в интерфейсе командной строки Mininet, чтобы просмотреть доступные линки: `net`.

Вывод этой команды показывает: - Хост `h1` подключён через свой сетевой интерфейс `h1-eth0` к коммутатору на интерфейсе `s1-eth1`. - Хост `h2` подключён

через свой сетевой интерфейс h2-eth0 к коммутатору на интерфейсе s1-eth2. - Коммутатор s1: - имеет петлевой интерфейс lo. - подключается к h1-eth0 через интерфейс s1-eth1. - подключается к h2-eth0 через интерфейс s1-eth2.

Mininet позволяет выполнять команды на конкретном устройстве. Чтобы выполнить команду для определенного узла, необходимо сначала указать устройство, а затем команду, например: `h1 ifconfig`.

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

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, используем 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` (рис. 2.13).

```

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 86:47:f7:b7:bb:48 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=2.14 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.324 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.065 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4082ms
rtt min/avg/max/mdev = 0.062/0.531/2.138/0.809 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 103.109 seconds

```

Рисунок 2.13: Проверка связности хостов

В терминале виртуальной машины mininet запустим MiniEdit: sudo ~/mininet/mininet/examples/miniedit.py.

Основные кнопки:

- Select: позволяет выбирать/перемещать устройства.
- Нажатие Del на клавиатуре после выбора устройства удаляет его из топологии.
- Host: позволяет добавить новый хост в топологию. После нажатия этой кнопки щелкните в любом месте пустого холста, чтобы вставить новый хост.
- Switch: позволяет добавить в топологию новый коммутатор. После нажатия этой кнопки щёлкните в любом месте пустого холста, чтобы вставить переключатель.
- Link: соединяет устройства в топологии. После нажатия этой кнопки щелкните устройство и перетащите его на второе устройство, с которым необходимо установить связь.
- Run: запускает эмуляцию. После проектирования и настройки топологии нажмите кнопку запуска.
- Stop: останавливает эмуляцию.

Добавим два хоста и один коммутатор, соединим хосты с коммутатором.

Настроим IP-адреса на хостах h1 и h2. Для этого удерживая правую кнопку мыши на устройстве выберем свойства. Для хоста h1 укажем IP-адрес 10.0.0.1/8, а для хоста h2 – 10.0.0.2/8.

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

Откроем терминал на хосте h1, удерживая правую кнопку мыши на хосте h1 и выбрав Terminal. Это действие позволит выполнять команды на хосте h1. Откроем терминал на хосте h2. На терминале хоста h1 введем команду ifconfig, чтобы отобразить назначенные ему IP-адреса. Интерфейс h1-eth0 на хосте h1 настроен с IP-адресом 10.0.0.1 и маской подсети 255.0.0.0. Повторим эти действия на хосте h2. Его интерфейс h2-eth0 настроен с IP-адресом 10.0.0.2 и маской подсети 255.0.0.0. Проверим соединение между хостами, введя в терминале хоста h1 команду ping 10.0.0.2. Для остановки теста нажмем Ctrl + c. Остановим эмуляцию, нажав кнопку Stop (рис. 2.14).

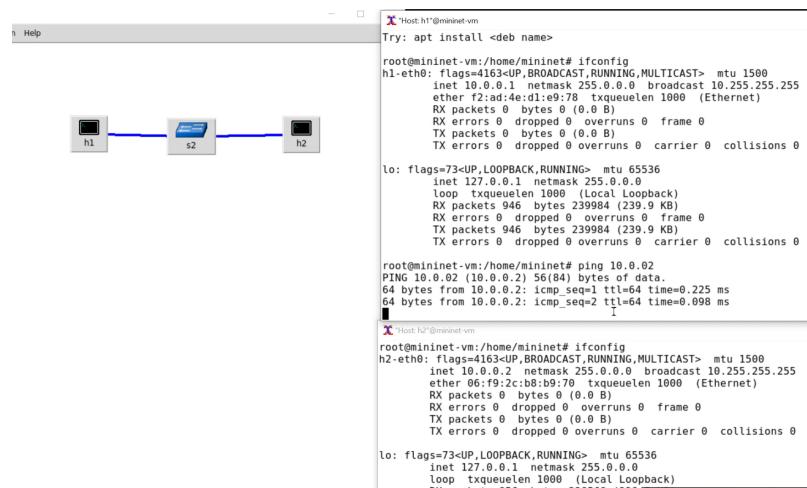


Рисунок 2.14: Проверка IP-адресов. Пинг

Ранее IP-адреса узлам h1 и h2 были назначены вручную. В качестве альтернативы можно полагаться на Mininet для автоматического назначения

IP-адресов.

Удалим назначенный вручную IP-адрес с хостов h1 и h2. В MiniEdit нажмем Edit Preferences . По умолчанию в поле базовые значения IP-адресов (IP Base) установлено 10.0.0.0/8. Изменим это значение на 15.0.0.0/8 (рис. 2.15).

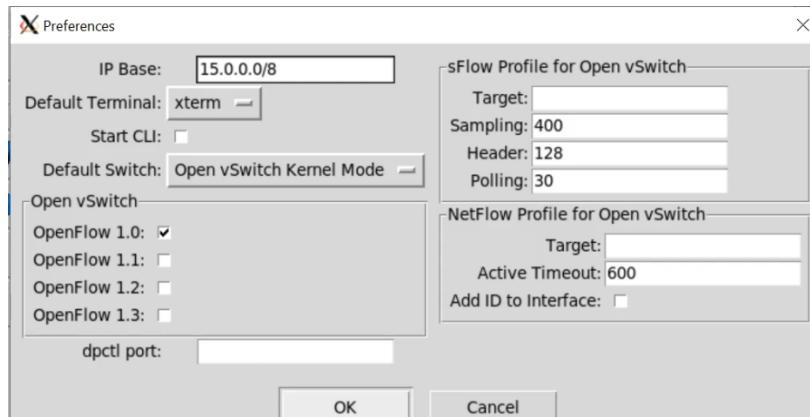


Рисунок 2.15: Смена базового IP-адреса

Запустим эмуляцию, нажав кнопку Run. Откроем терминал на хосте h1, удерживая правую кнопку мыши на хосте h1 и выбрав Terminal. Чтобы отобразить IP-адреса, назначенные хосту h1, введем команду ifconfig. Интерфейс h1-eth0 на узле h1 теперь имеет IP-адрес 15.0.0.1 и маску подсети 255.0.0.0. Проверим IP-адрес, назначенный хосту h2. Соответствующий интерфейс h2-eth0 на хосте h2 имеет IP-адрес 15.0.0.2 и маску подсети 255.0.0.0(рис. 2.16).

```

Terminal "Host: h1" @ 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 c6:02:03:05:53:9e 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 854 bytes 228672 (228.6 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 854 bytes 228672 (228.6 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# 

Terminal "Host: h2" @ 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 22:39:cd:55:10:26 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 228412 (228.4 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 853 bytes 228412 (228.4 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# 

```

Рисунок 2.16: Просмотр IP-адреса на h1

Остановим эмуляцию, нажав кнопку Stop.

В домашнем каталоге виртуальной машины mininet создадим каталог для работы с проектами mininet: `mkdir ~ / work`. Для сохранения топологии сети в файл нажмем в MiniEdit File Save. Укажем имя для топологии и сохраним на своем компьютере. После сохранения проекта поменяем права доступа к файлам в каталоге проекта: `sudo chown -R mininet:mininet ~ / work`. Для загрузки топологии в MiniEdit нажмем File Open (рис. 2.17).

```

[mininet@mininet-vm:~]$ mkdir ~ / work
[mininet@mininet-vm:~]$ sudo ~/mininet/mininet/examples/miniedit.py
topo=none
[mininet@mininet-vm:~]$ sudo chown -R mininet:mininet ~ / work

```

Рисунок 2.17: Сохранение топологии

Завершим соединение с виртуальной машиной mininet и выключим её.

3 Выводы

В ходе выполнения лабораторной работы я развернула mininet в системе виртуализации VirtualBox и ознакомилась с основными командами для работы с Mininet через командную строку и через графический интерфейс.