

Лабораторная работа № 1. Введение в Mininet

Абакумова Олеся Максимовна, НФИбд-02-22

Содержание

1	Цель работы	6
2	Теоретическое введение	7
3	Выполнение лабораторной работы	8
3.1	Настройка стенда виртуальной машины Mininet	8
3.2	Настройка параметров XTerm	16
3.3	Настройка соединения X11 для суперпользователя	17
3.4	Настройка доступа к Интернет	18
3.5	Обновление версии Mininet	19
3.6	Основы работы в Mininet	20
3.6.1	Работа с Mininet с помощью командной строки	20
3.6.2	Построение и эмуляция сети в Mininet с использованием графического интерфейса	24
4	Выводы	32

Список иллюстраций

3.1	Импорт конфигураций	8
3.2	Импорт конфигураций	9
3.3	В системе отсутствуют неисправности	9
3.4	Первый сетевой адаптер NAT	10
3.5	Второй сетевой адаптер(виртуальный адаптер хоста)	10
3.6	Информация о виртуальной машине	11
3.7	Пингование виртуальной машины(успешно)	11
3.8	Подключение к виртуальной машине	12
3.9	Установленные putty и VcXsrv Windows X Server	12
3.10	Выбор опций	13
3.11	Выбор опций	13
3.12	Подключение в putty	14
3.13	Подключение в putty	15
3.14	Добавление опции перенаправления X11	16
3.15	Выбран системный моноширинный шрифт, кегль шрифта – 12 пунктов	17
3.16	Копирование значения куки и запуск xterm	17
3.17	Запуск Wireshark	18
3.18	Вывод команды ifconfig	18
3.19	Активен адрес NAT 10.0.2.15	19
3.20	Внесение изменений в 01-netcfg.yaml	19
3.21	Установка новой версии Mininet	20
3.22	Версия Mininet	20
3.23	Запуск минимальной топологии и отображение списка команд ин- терфейса	21
3.24	Доступные узлы и отображение связи между устройствами	22
3.25	Конфигурация h2 и s1	23
3.26	Пингование 10.0.0.2	24
3.27	Запуск MiniEdit	24
3.28	Добавление двух хостов и одного коммутатора в рабочую область	26
3.29	Настройка IP-адреса на h1	27
3.30	Настройка IP-адреса на h2	28
3.31	Запуск эмуляции	29
3.32	Открытие терминала на h1 и h2	29
3.33	Отображение, назначенных IP-адресов для h1 и h2	30
3.34	Пингование 10.0.0.1	30
3.35	Настройка автоматического назначения адресов на h1 и h2	30

3.36 Успешное автоматическое назначение IP-адресов	31
3.37 Сохранение проекта	31

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 Теоретическое введение

Mininet (<http://mininet.org/>) – это виртуальная среда, которая позволяет разрабатывать и тестировать сетевые инструменты и протоколы. В сетях Mininet работают реальные сетевые приложения Unix/Linux, а также реальное ядро Linux и сетевой стек.

3 Выполнение лабораторной работы

3.1 Настройка стенда виртуальной машины Mininet

Для начала сделаем импорт конфигураций предварительно установив необходимый файл (рис. 3.1):

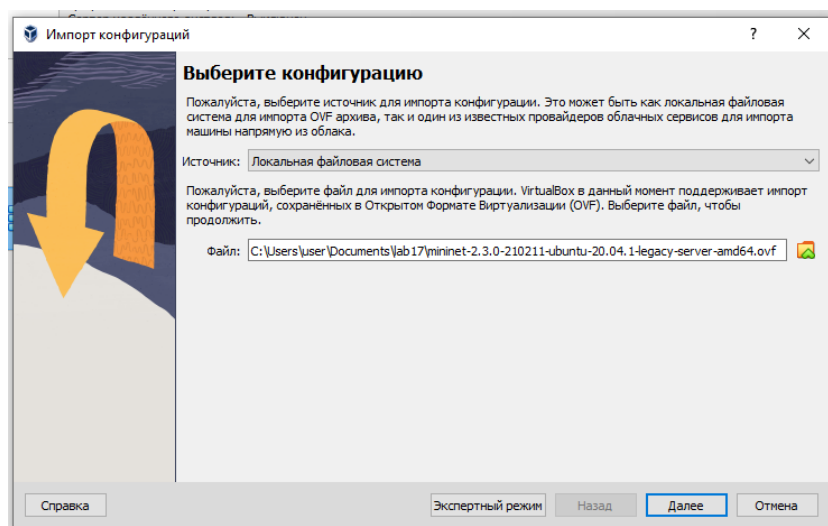


Рис. 3.1: Импорт конфигураций

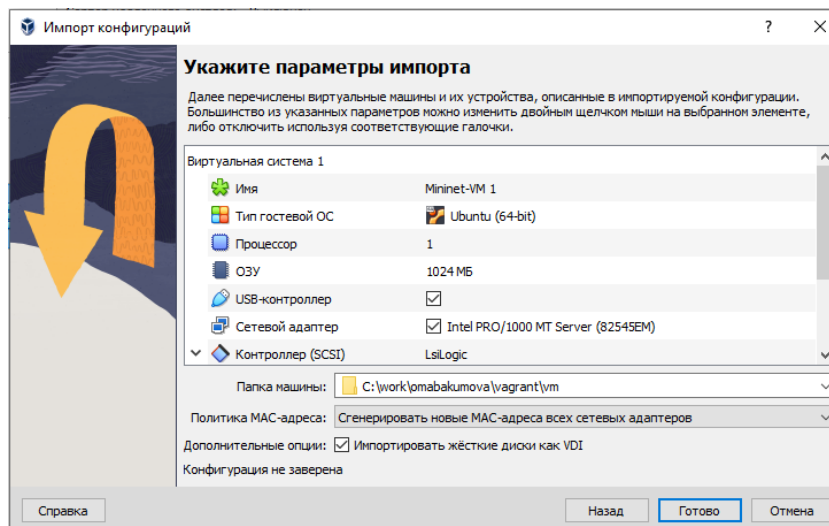


Рис. 3.2: Импорт конфигураций

Теперь необходимо внести некоторые настройки в виртуальную машину (рис. 3.3):

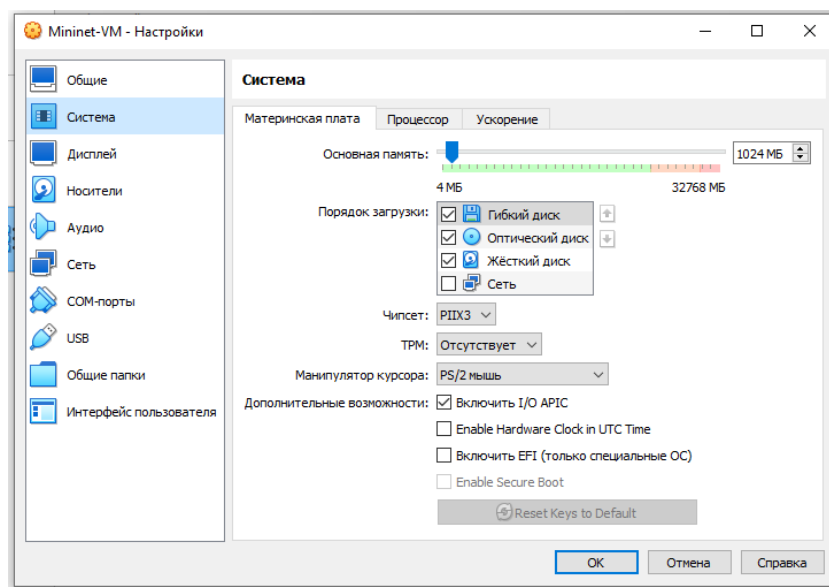


Рис. 3.3: В системе отсутствуют неисправности

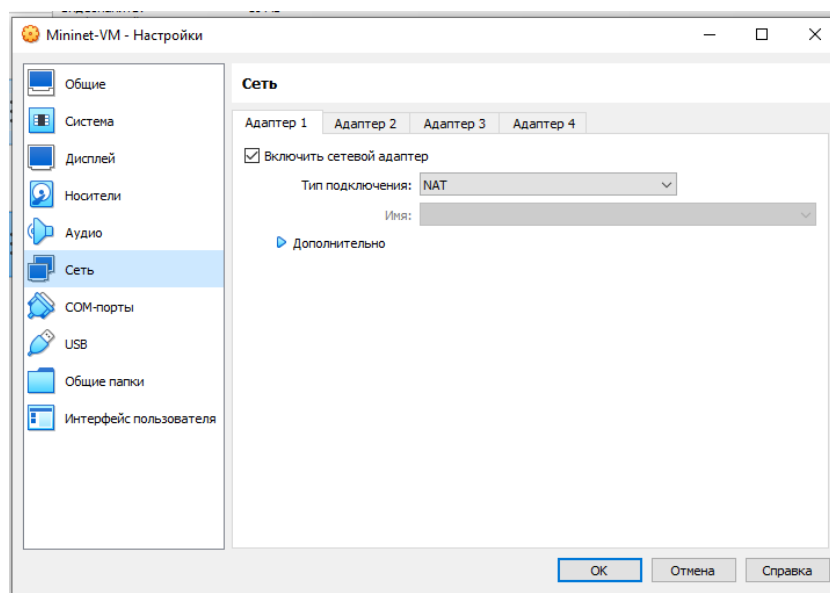


Рис. 3.4: Первый сетевой адаптер NAT

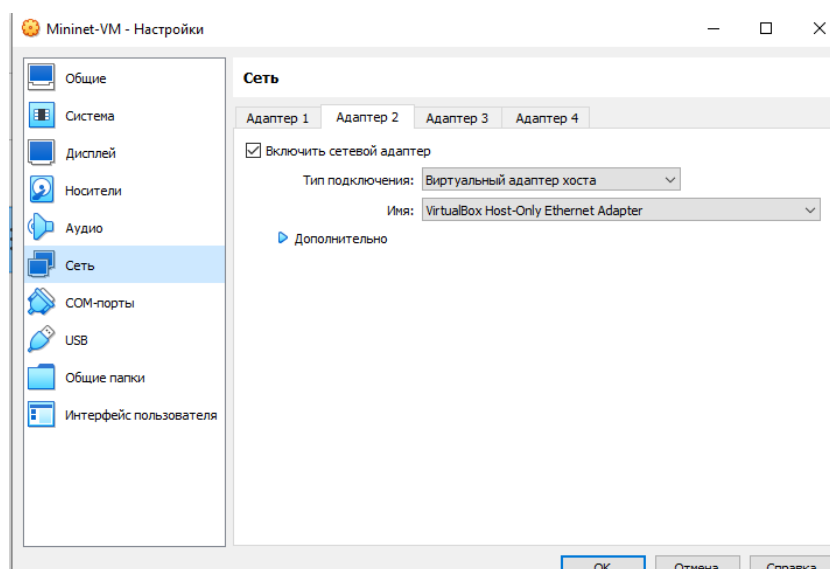


Рис. 3.5: Второй сетевой адаптер(виртуальный адаптер хоста)

Запустим виртуальную машину и выведем информацию о ней с помощью 'ifconfig' (рис. 3.6):

```
Mininet-VM [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
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 09:00:50 PDT 2025 from 192.168.56.1 on pts/1
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 08:00:27:40:a4:3f  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

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 08:00:27:19:7d:07  txqueuelen 1000  (Ethernet)
    RX packets 116  bytes 12211 (12.2 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 119  bytes 11194 (11.1 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 46  bytes 3700 (3.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 46  bytes 3700 (3.7 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$
```

Рис. 3.6: Информация о виртуальной машине

Зная статический IP-адрес, проверим его доступность с хостовой машины(из под Windows)(рис. 3.7):

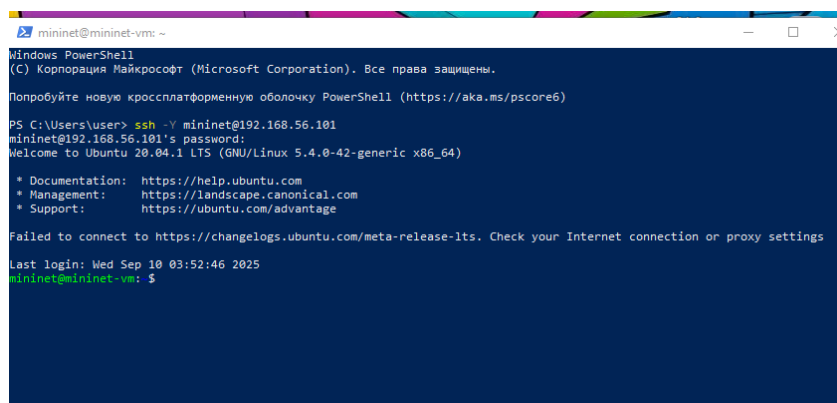
```
connection to 192.168.56.101 closed.
PS C:\Users\user> ping 192.168.56.101

Обмен пакетами с 192.168.56.101 по 32 байтами данных:
Ответ от 192.168.56.101: число байт=32 время<1мс TTL=64
Ответ от 192.168.56.101: число байт=32 время<1мс TTL=64
Ответ от 192.168.56.101: число байт=32 время<1мс TTL=64
Ответ от 192.168.56.101: число байт=32 время<1мс TTL=64

Статистика Ping для 192.168.56.101:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потеря)
    Приблизительное время приема-передачи в мс:
        Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
PS C:\Users\user>
```

Рис. 3.7: Пингование виртуальной машины(успешно)

Теперь подключимся по SSH (рис. 3.8):



```
mininet@mininet-vm: ~
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\User> ssh -Y mininet@192.168.56.101
mininet@192.168.56.101's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

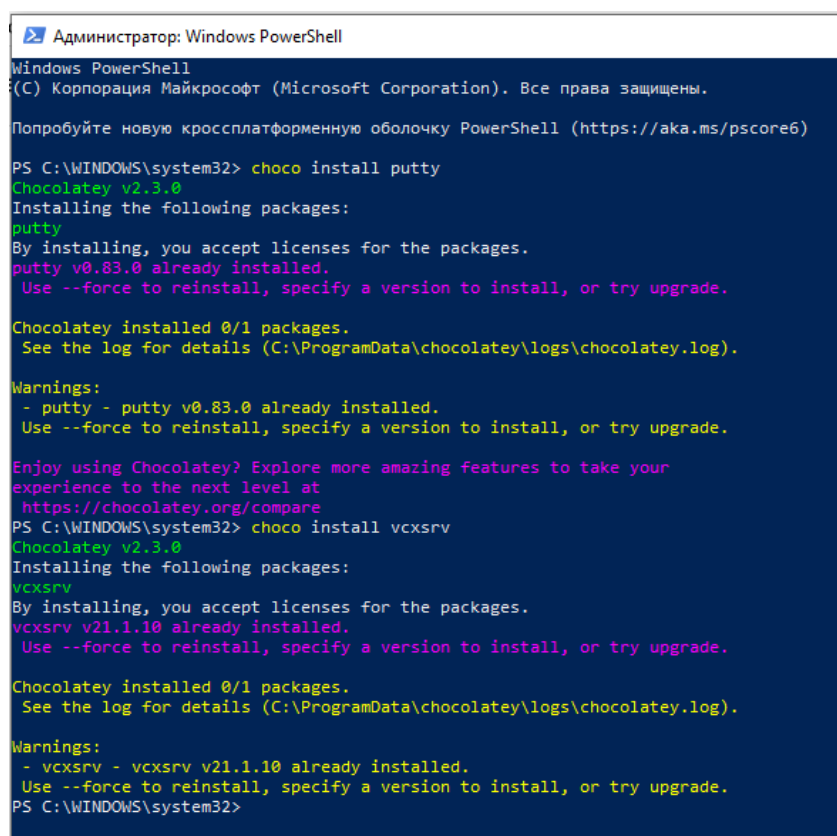
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

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

Last login: Wed Sep 10 03:52:46 2025
mininet@mininet-vm: $
```

Рис. 3.8: Подключение к виртуальной машине

Далее нам необходимо установить следующие утилит для работы из под Windows (рис. 3.9):



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

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\WINDOWS\system32> choco install putty
Chocolatey v2.3.0
Installing the following packages:
putty
By installing, you accept licenses for the packages.
putty v0.83.0 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 v0.83.0 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Enjoy using Chocolatey? Explore more amazing features to take your
experience to the next level at
https://chocolatey.org/compare
PS C:\WINDOWS\system32> choco install vcxsrv
Chocolatey v2.3.0
Installing the following packages:
vcxsrv
By installing, you accept licenses for the packages.
vcxsrv v21.1.10 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:
- vcxsrv - vcxsrv v21.1.10 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
PS C:\WINDOWS\system32>
```

Рис. 3.9: Установленные putty и VcXsrv Windows X Server

Перейдем к запуску Xserver (рис. 3.10):

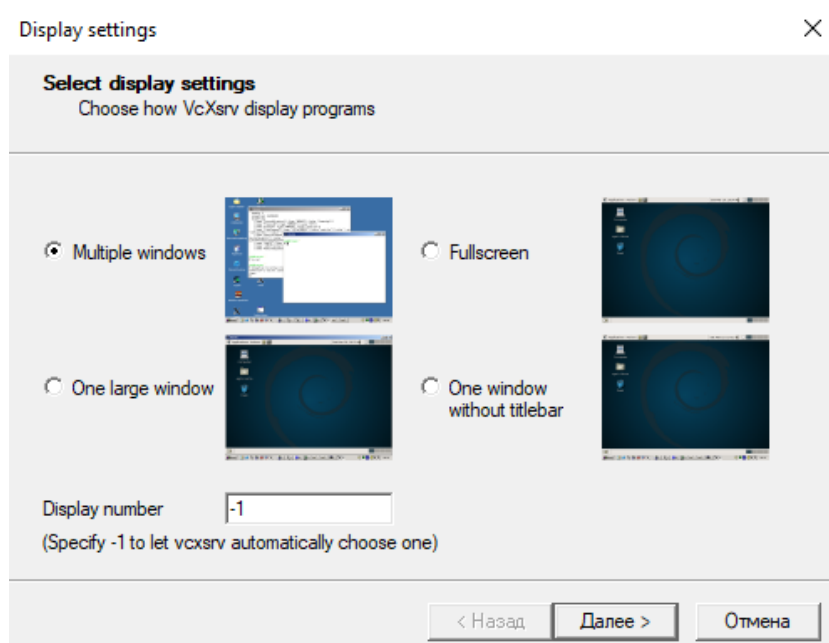


Рис. 3.10: Выбор опций

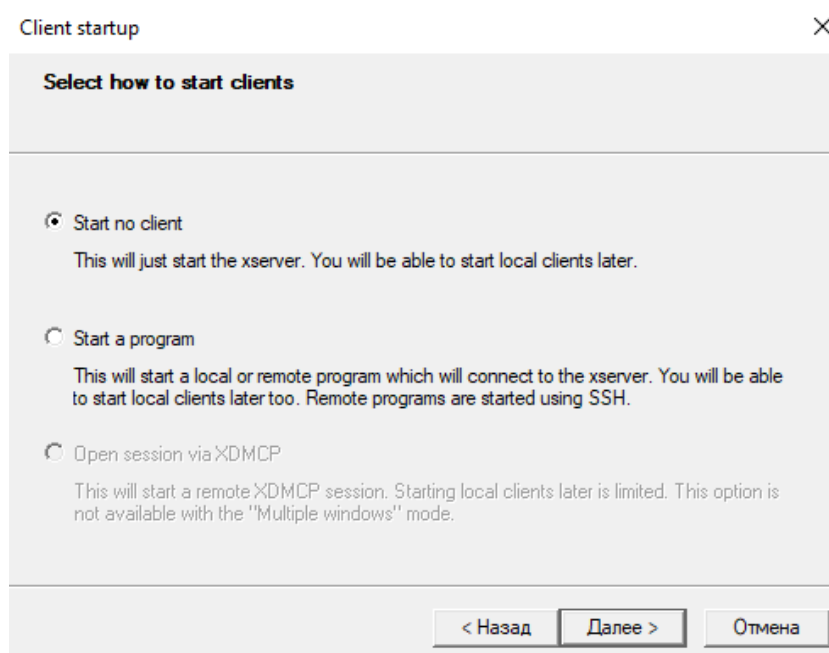


Рис. 3.11: Выбор опций

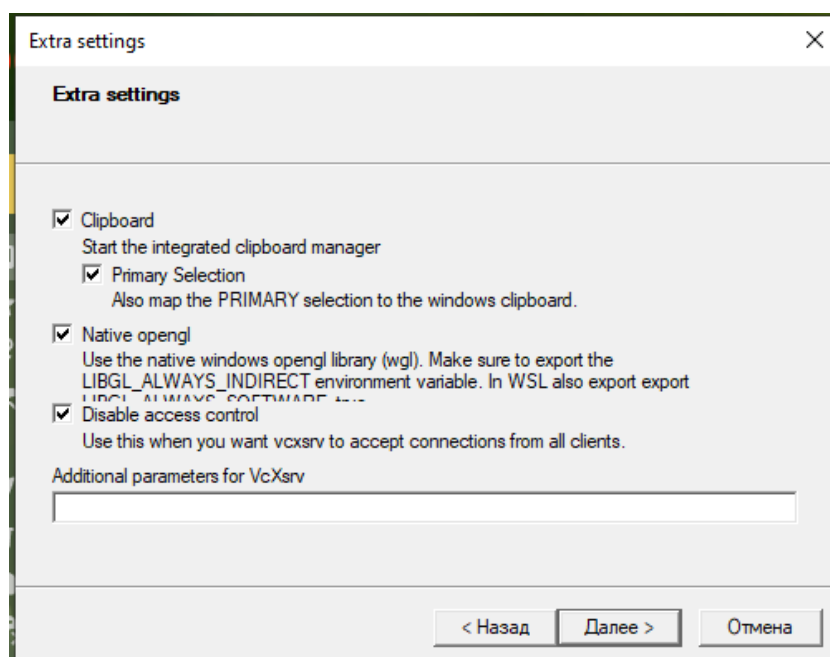


Рис. 3.12: Подключение в putty

Перейдем к putty (рис. 3.13):

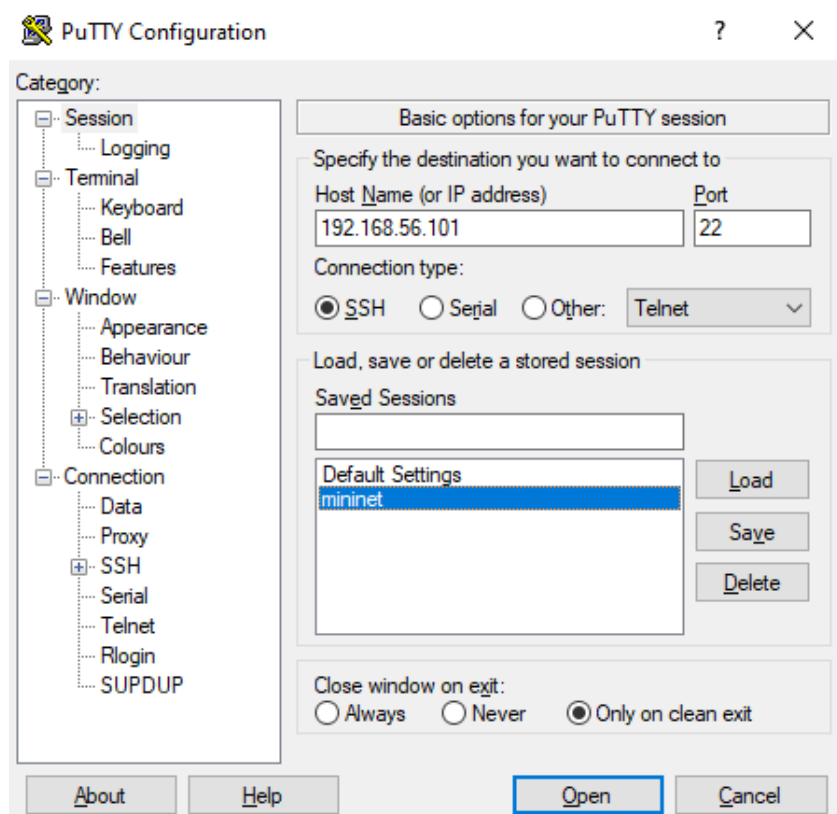


Рис. 3.13: Подключение в putty

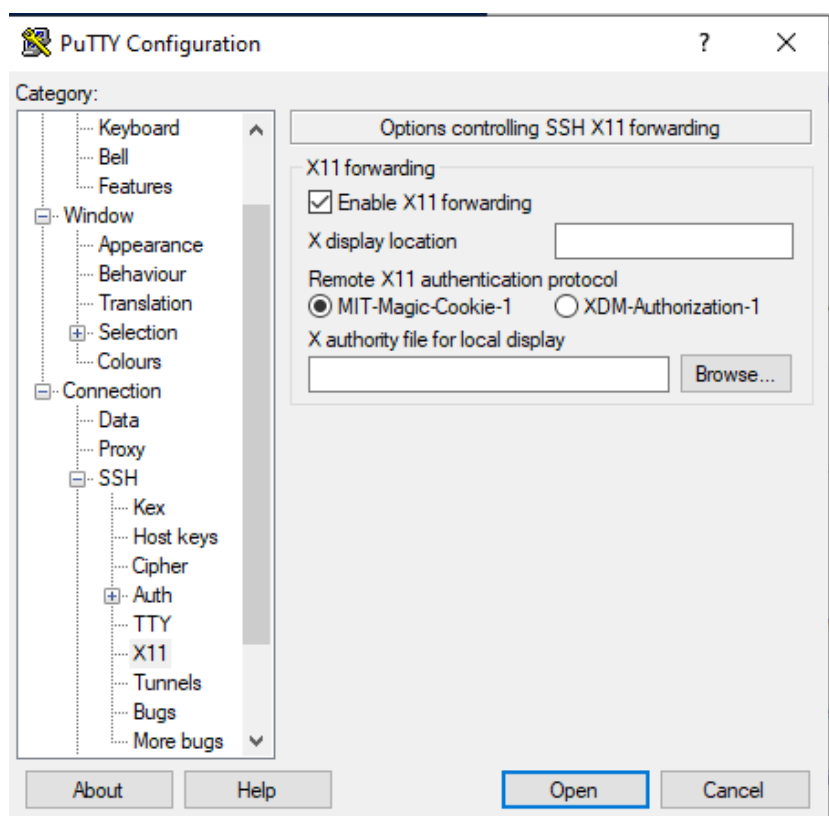


Рис. 3.14: Добавление опции перенаправления X11

3.2 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` (рис. 3.15):


```
/etc/X11~s/XTerm [----] 18 L:[245+21 266/266] *(10376/10376b) <EOF> [*][X]
! Alternatively,
! on2Clicks: regex [[[:alpha:]]+://([[:alnum:]]!#+,./=?@_~)|(%[[:xdigit:]])[[:xdig
!
! 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
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDm10Quit
```

Рис. 3.15: Выбран системный моноширинный шрифт, кегль шрифта – 12 пунктов

3.3 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. Скопируем значение куки (MIT magic cookie) пользователя mininet в файл для пользователя root (рис. 3.16):

```
mininet@mininet-vm: ~
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 494339ad258705a93254e75e0839d5b2
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 b2285497a8824753c8f3991ee30a6ab9
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 494339ad25
705a93254e75e0839d5b2
root@mininet-vm:~# logout
mininet@mininet-vm:~$ xterm
mininet@mininet-vm:~$ wireshark
libGL error: No matching fbConfigs or visuals found
libGL error: failed to load driver: swrast
07:13:46.537 Warn Could not compile "of" in colorfilters file "/home/m
mininet/.wireshark/colorfilters".
"of" is neither a field nor a protocol name.
07:13:46.539 Warn Could not compile "Checksum Errors" in colorfilters
file "/home/mininet/.wireshark/colorfilters".
Neither "cdp.checksum_bad" nor "1" are field or protocol names.
mininet@mininet-vm:~$ sudo mcedit /etc/X11/app-defaults/XTerm
mininet@mininet-vm:~$
mininet@mininet-vm:~$
mininet@mininet-vm:~$
```

Рис. 3.16: Копирование значения куки и запуск xterm

У меня на скриншотах возможна настройка выглядит не совсем как ожидается так, как я проделывала это не в первый раз. Как можно заметить после внесения этого изменения в качестве теста мы можем открыть различные утилиты, как из под рута, так и из под пользователя (рис. 3.17):

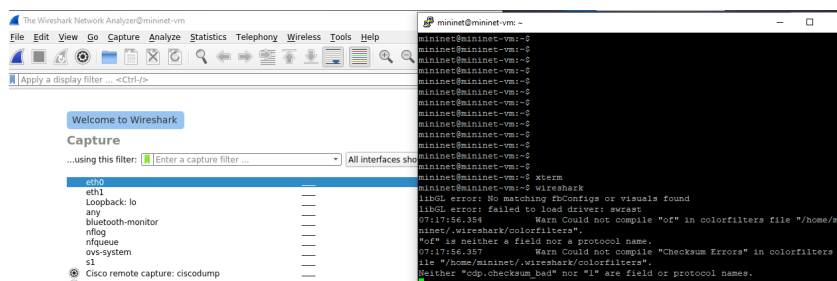


Рис. 3.17: Запуск Wireshark

3.4 Настройка доступа к Интернет

Теперь перейдем к настройке доступа к Интернету. До настройки доступа мы имеем следующее (рис. 3.18):

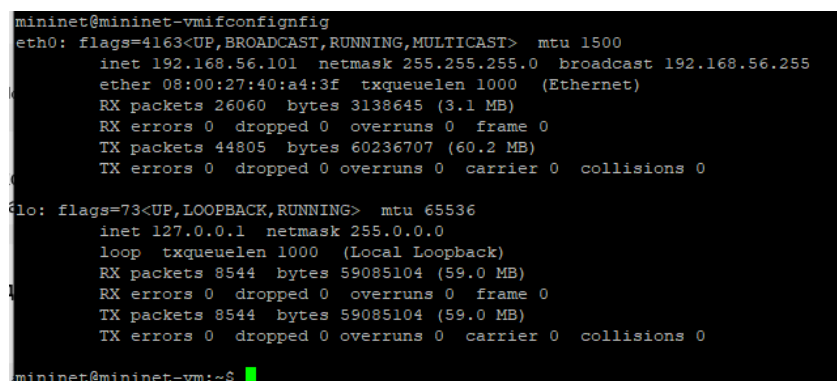


Рис. 3.18: Вывод команды ifconfig

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 активируем второй интерфейс, набрав в командной строке: 'sudo dhclient eth1' и 'ifconfig' (рис. 3.19):

```

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 08:00:27:40:a4:3f txqueuelen 1000 (Ethernet)
    RX packets 26072 bytes 3139671 (3.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 44814 bytes 60238729 (60.2 MB)
    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 08:00:27:19:7d:07 txqueuelen 1000 (Ethernet)
    RX packets 392 bytes 38403 (38.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 407 bytes 37172 (37.1 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 8558 bytes 59085980 (59.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8558 bytes 59085980 (59.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$

```

Рис. 3.19: Активен адрес NAT 10.0.2.15

Для удобства дальнейшей работы установим mc с помощью команды ‘sudo apt install mc’ и 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 minine (рис. 3.20):

```

mininet@mininet-vm: ~
/etc/net~cfg.yaml [----] 0 L:[ 1+ 2 3/ 11] *(110 / 220b) 110 0x06E [*][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

```

Рис. 3.20: Внесение изменений в 01-netcfg.yaml

3.5 Обновление версии Mininet

Необходимо обновить версию Mininet и произвести установку (рис. 3.21):

```

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% (140/140), done.
remote: Total 10388 (delta 129), reused 174 (delta 92), pack-reused 10154 (from 1)
Receiving objects: 100% (10388/10388), 3.36 MiB | 9.07 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 2>&1\" mnexec.c -o mnexec

```

Рис. 3.21: Установка новой версии Mininet

Проверить версию можно следующим образом (рис. 3.22):

```

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

```

Рис. 3.22: Версия Mininet

3.6 Основы работы в Mininet

3.6.1 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' (рис. 3.23):

```

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>

```

Рис. 3.23: Запуск минимальной топологии и отображение списка команд интерфейса

Для отображения доступных узлов введем: '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' 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 (рис. 3.24):

```
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> 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 ba:9d:f6:0b:4c:5d 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>
```

Рис. 3.24: Доступные узлы и отображение связи между устройствами

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 (рис. 3.24).

Посмотрим конфигурацию всех узлов (рис. 3.25):

```

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 c6:9b:fb:45:f4:ae 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> 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 08:00:27:40:a4:3f txqueuelen 1000 (Ethernet)
    RX packets 26268 bytes 3162827 (3.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 44989 bytes 60265239 (60.2 MB)
    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 08:00:27:19:7d:07 txqueuelen 1000 (Ethernet)
    RX packets 435 bytes 42123 (42.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 451 bytes 40982 (40.9 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 8755 bytes 59101804 (59.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8755 bytes 59101804 (59.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 92:a8:27:fe:29:55 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 3a:af:d8:71:bb:d5 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

mininet>

```

Рис. 3.25: Конфигурация h2 и s1

Эти записи выполняют команду `ifconfig` на хосте h2 и показывает интерфейсы хоста h2 – хост h2 имеет интерфейс h2-eth0, настроенный с IP-адресом 10.0.0.2, и другой интерфейс lo, настроенный с IP-адресом 127.0.0.1. Также выведена информация о s1 с IP-адресом 192.168.56.101.

Проверим связность. 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, мы можем использовать команду ping. Команда ping работает, отправляя сообщения эхо-запроса протокола управляющих сообщений Интернета (ICMP) на удалённый компьютер и ожидая ответа (рис. 3.26):

```
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=3.28 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.248 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.055 ms
^Z
[1]+  Stopped                  sudo mn
mininet@mininet-vm:~$ exit
logout
There are stopped jobs.
mininet@mininet-vm:~$
```

Рис. 3.26: Пингование 10.0.0.2

Мы проверили соединение между хостами h1 и h2. Для остановки эмуляции достаточно ввести: 'exit'.

3.6.2 Построение и эмуляция сети в Mininet с использованием графического интерфейса

В терминале виртуальной машины mininet запустим MiniEdit (рис. 3.27):

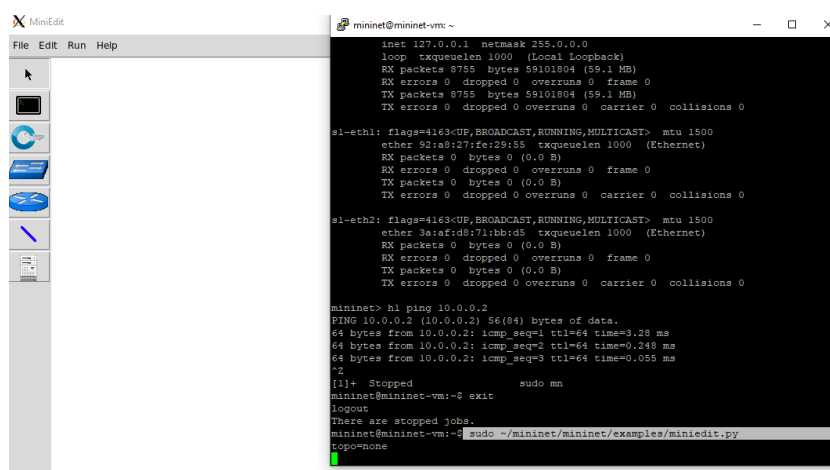


Рис. 3.27: Запуск MiniEdit

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 щелчком в любом месте пустого холста, чтобы вставить новый хост.
- Switch: позволяет добавить в топологию новый коммутатор. После нажатия этой кнопки щелчком в любом месте пустого холста, чтобы вставить переключатель.
- Link: соединяет устройства в топологии. После нажатия этой кнопки щелкните устройство и перетащим его на второе устройство, с которым необходимо установить связь.
- Run: запускает эмуляцию. После проектирования и настройки топологии нажмем кнопку запуска.
- Stop: останавливает эмуляцию.

В качестве демонстрации добавим два хоста и один коммутатор, соединим хосты с коммутатором (рис. 3.28):

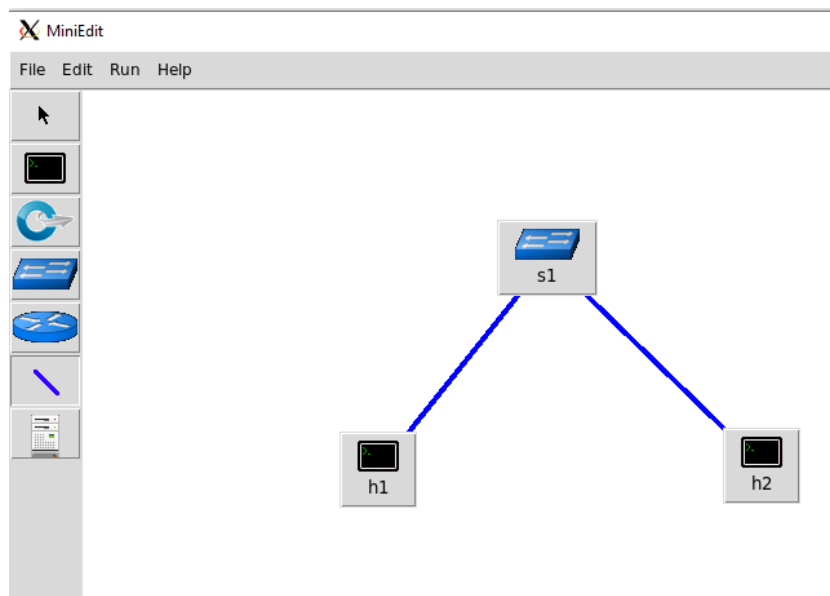


Рис. 3.28: Добавление двух хостов и одного коммутатора в рабочую область

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 (рис. 3.29):

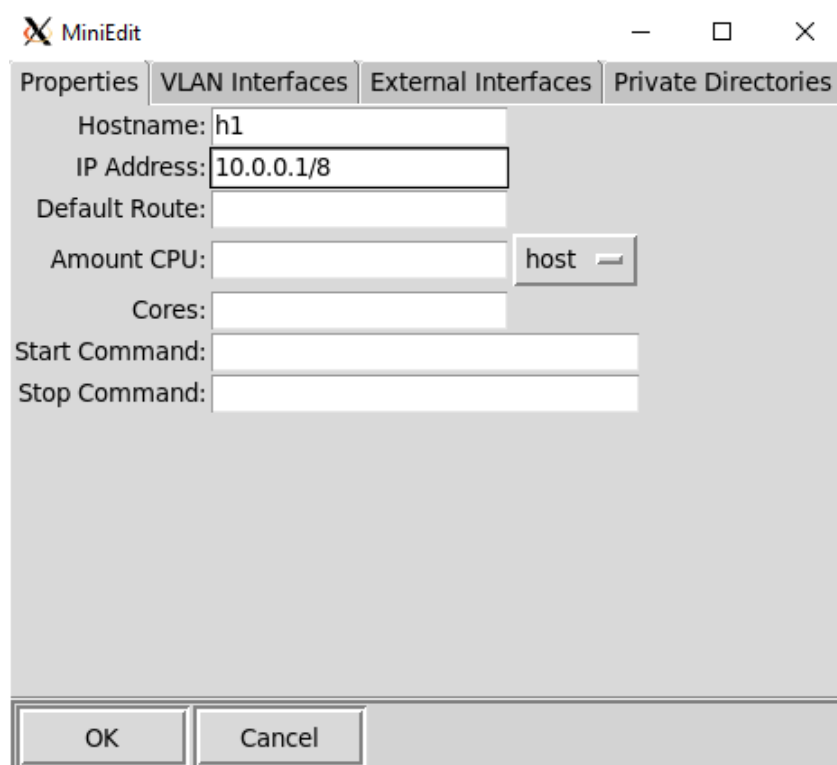


Рис. 3.29: Настройка IP-адреса на h1

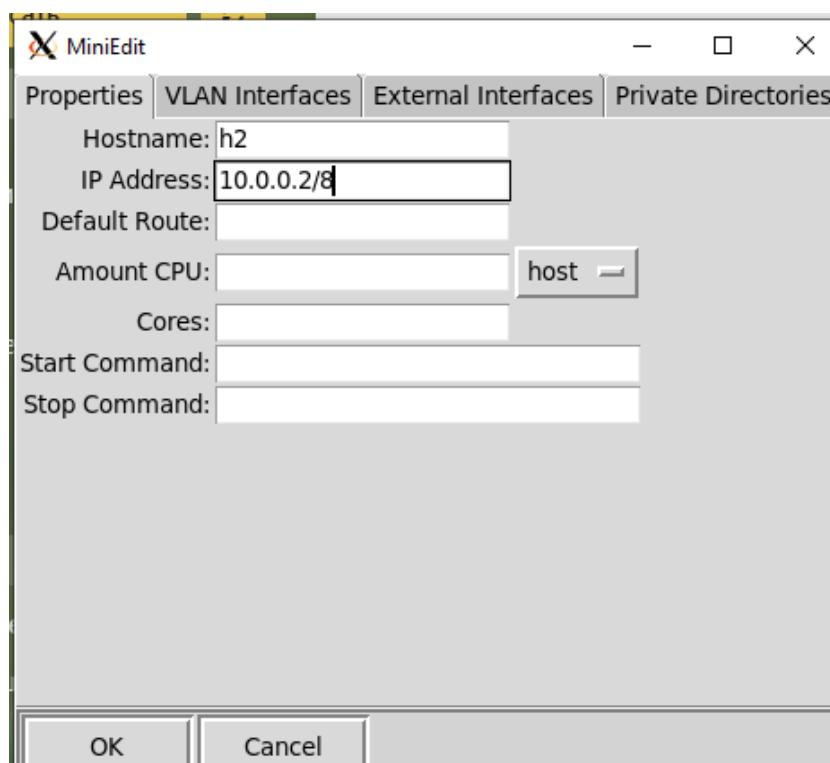


Рис. 3.30: Настройка IP-адреса на h2

Проверим связность. 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 (рис. 3.31):

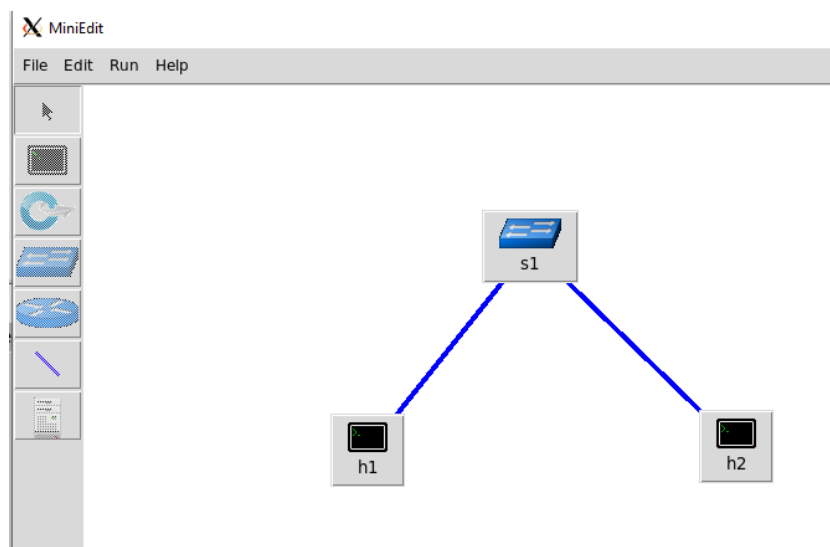


Рис. 3.31: Запуск эмуляции

Откроем терминал на обоих хостах (рис. 3.32):

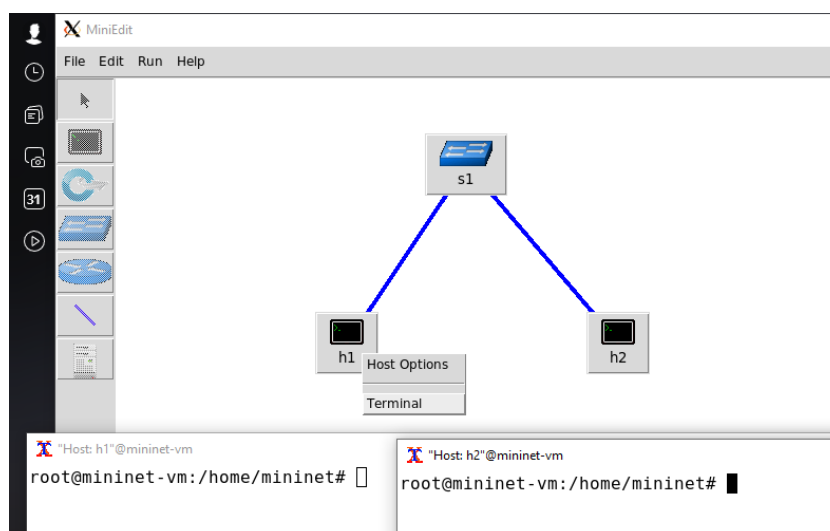


Рис. 3.32: Открытие терминала на h1 и h2

На терминале хоста h1 введем команду `ifconfig`, чтобы отобразить назначенные 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`. Повторим 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` (рис. 3.33);

```
Host h1@mininet-vn
root@mininet-vn:/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 12:8c:37:c5:c3:7c 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 1374 bytes 271876 (271.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1374 bytes 271876 (271.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vn:/home/mininet#

Host h2@mininet-vn
root@mininet-vn:/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 86:88:69:c8:b0:f5 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 1620 bytes 299436 (299.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1620 bytes 299436 (299.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 3.33: Отображение, назначенных IP-адресов для h1 и h2

Проверим соединение между хостами (рис. 3.34):

```
root@mininet-vn:/home/mininet# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
 64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.027 ms
 64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.041 ms
 64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.043 ms
^Z
[1]+  Stopped                  ping 10.0.0.1
root@mininet-vn:/home/mininet#
```

Рис. 3.34: Пингование 10.0.0.1

Остановить эмуляцию можно просто нажав ‘stop’.

Ранее IP-адреса узлов h1 и h2 были назначены вручную. В качестве альтернативы можно полагаться на Mininet для автоматического назначения IP-адресов. Предварительно удалив назначенные вручную IP-адреса хостов h1 и h2 перейдем к автоматическому назначению (рис. 3.35):

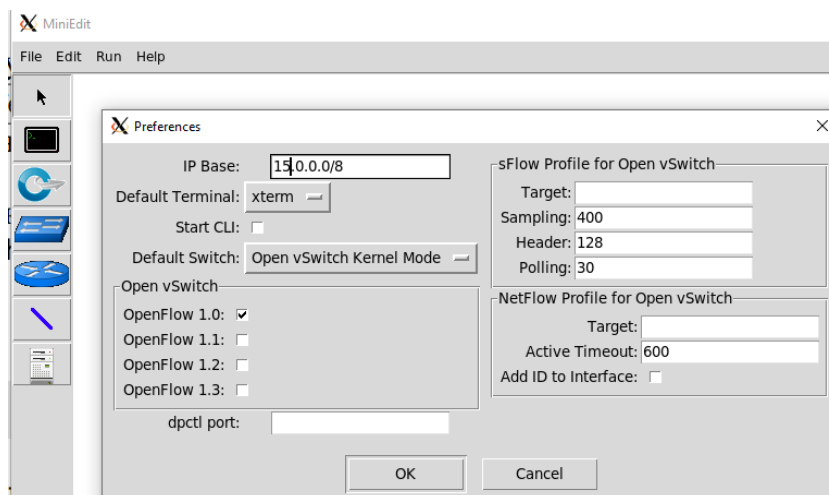


Рис. 3.35: Настройка автоматического назначения адресов на h1 и h2

Удостовериться в назначении можно просто выполнив ‘ifconfig’ (рис. 3.36):

```
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 9a:37:d0:ba:d3:66 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 228120 (228.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 854 bytes 228120 (228.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

```
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 a6:30:02:66:f7:c2 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 228088 (228.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 854 bytes 228088 (228.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 3.36: Успешное автоматическое назначение IP-адресов

Сохранить и загрузить топологию в Mininet достаточно просто. В домашнем каталоге виртуальной машины mininet создадим каталог ‘work’ для работы с проектами mininet. Также выполним команду ‘sudo chown -R mininet:mininet ~/work’, чтобы поменять права доступа к файлам в каталоге проекта (рис. 3.37):

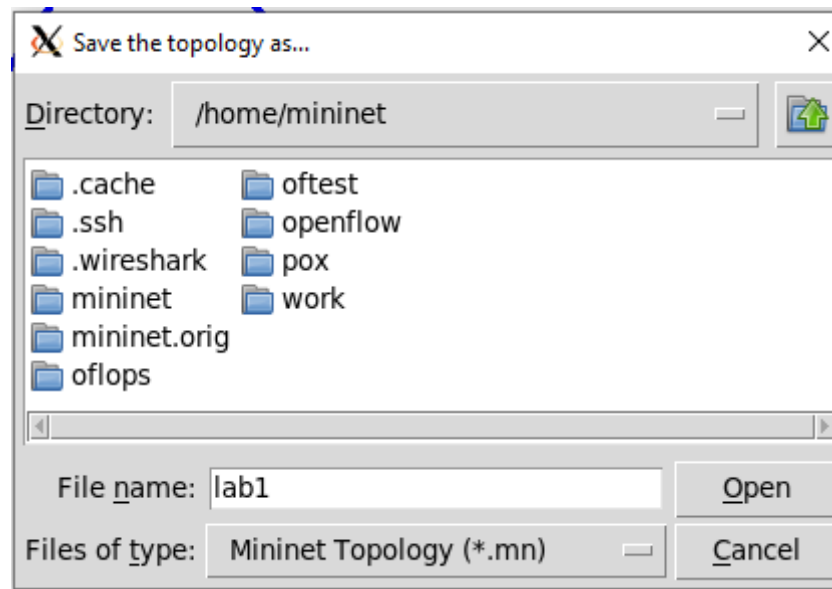


Рис. 3.37: Сохранение проекта

4 Выводы

В результате выполнения данной лабораторной работы я развёрнула mininet в системе виртуализации VirtualBox, а также ознакомилась с основными командами для работы с Mininet через командную строку и через графический интерфейс.