

# Лабораторная работа № 1. Введение в Mininet

---

Абакумова О. М.

Российский университет дружбы народов, Москва, Россия

## Информация

---

# Докладчик

---

- Абакумова Олеся Максимовна
- Студентка
- Российский университет дружбы народов
- 1132220832@pfur.ru
- <https://github.com/omabakumova>



## Цель работы

---

## 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.

## Теоретическое введение

---

## 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.

## Задания

---

## Задания

---

- Произвести настройку виртуальной машины Mininet
- Изучить основы работы в Mininet

## Выполнение лабораторной работы

---

# Настройка стенда виртуальной машины Mininet

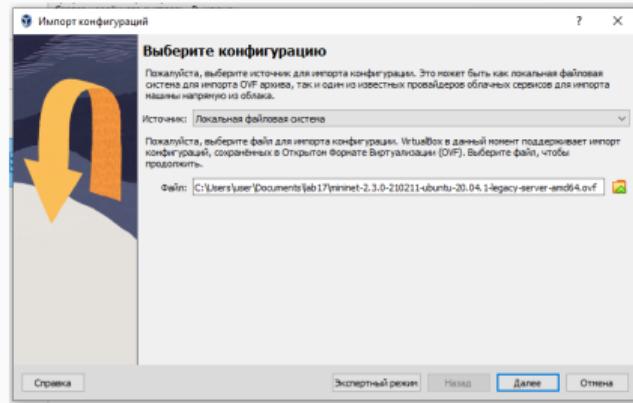


Рис. 1: Импорт конфигураций

# Настройка стенда виртуальной машины Mininet

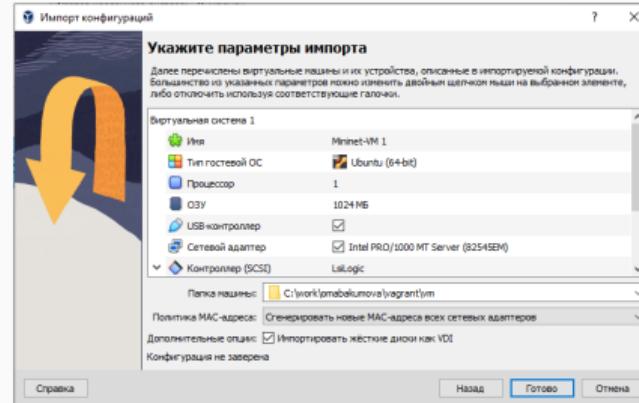


Рис. 2: Импорт конфигураций

# Настройка стенда виртуальной машины Mininet

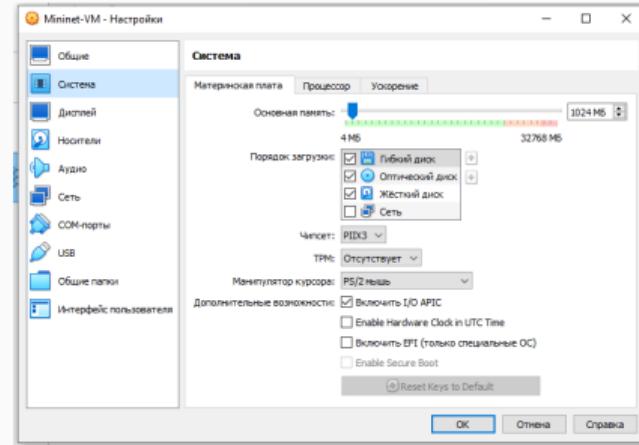


Рис. 3: В системе отсутствуют неисправности

# Настройка стенда виртуальной машины Mininet

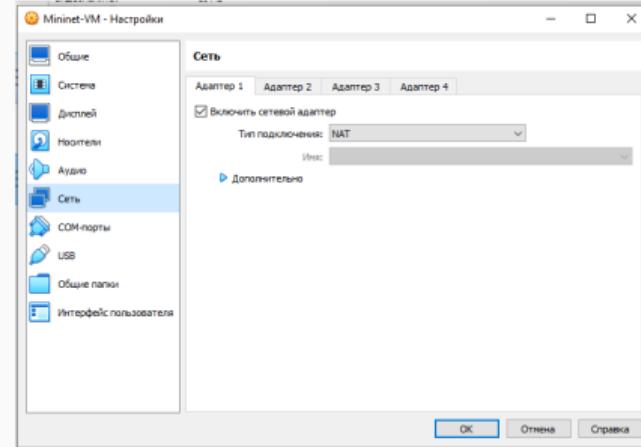


Рис. 4: Первый сетевой адаптер NAT

# Настройка стенда виртуальной машины Mininet

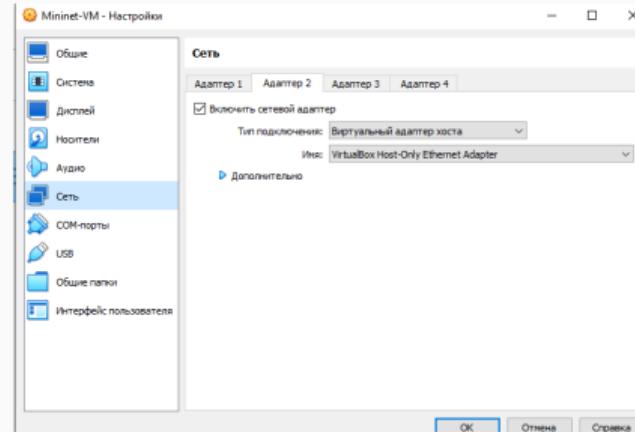
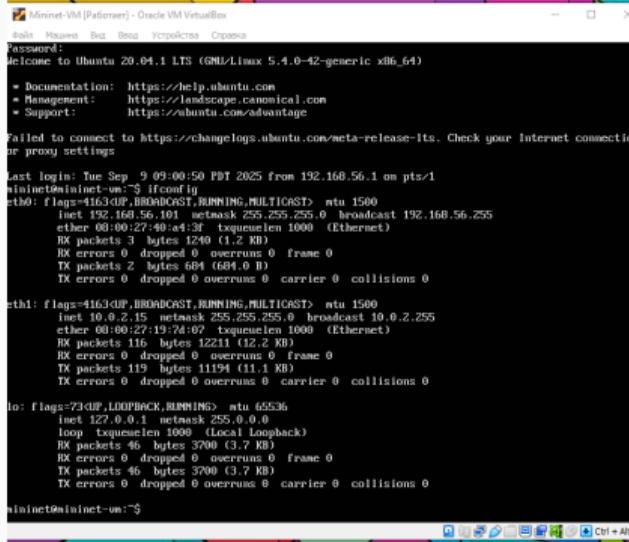


Рис. 5: Второй сетевой адаптер(виртуальный адаптер хоста)

# Настройка стенда виртуальной машины Mininet



```
Mininet-VM [Pablosser] - Oracle VM VirtualBox
File Edit View 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=4163UP,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:a1:3f 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=4163UP,BROADCAST,RUNNING,MULTICAST mtu 1500
      inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
          ether 08:00:27:19:74:07 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=73UP,LOOPBACK,BROADCAST,RUNNING mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
          loop txqueuelen 1000 (Local Loopback)
            RX packets 46 bytes 3906 (3.7 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 46 bytes 3906 (3.7 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
mininet@mininet-vm:~$
```

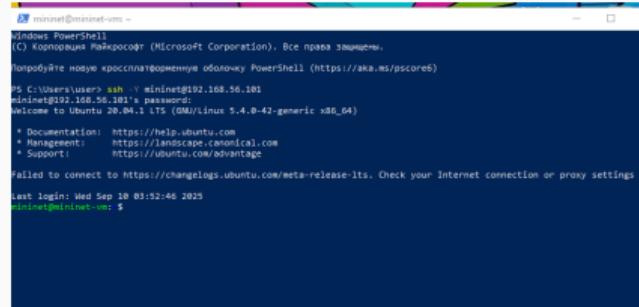
Рис. 6: Информация о виртуальной машине

# Настройка стенда виртуальной машины Mininet

```
connection to 192.168.56.101 closed.  
PS C:\Users\user> ping 192.168.56.101  
  
Обмен пакетами с 192.168.56.101 по с 32 байтами данных:  
Ответ от 192.168.56.101: число байт=32 время<1ms TTL=64  
  
Статистика Ping для 192.168.56.101:  
    Пакетов: отправлено = 4, получено = 4, потеряно = 0  
        (0% потерян)  
Приблизительное время приема-передачи в мс:  
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек  
PS C:\Users\user>
```

Рис. 7: Пингование виртуальной машины(успешно)

# Настройка стенда виртуальной машины Mininet



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

Установите новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscores)

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)

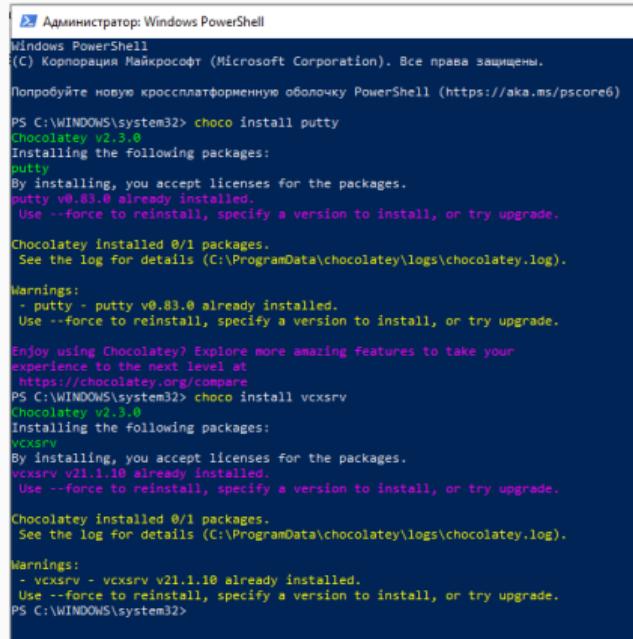
 * Documentation: https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/advantage

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

Last login: Wed Sep 18 03:52:46 2025
mininet@mininet-vm: $
```

Рис. 8: Подключение к виртуальной машине

# Настройка стенда виртуальной машины Mininet



```
Administrator: 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 vcxsvr
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:
- vcxsvr - vcxsvr v21.1.10 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
PS C:\WINDOWS\system32>
```

Рис. 9: Установленные putty и VcXsrv Windows X Server

# Настройка стенда виртуальной машины Mininet

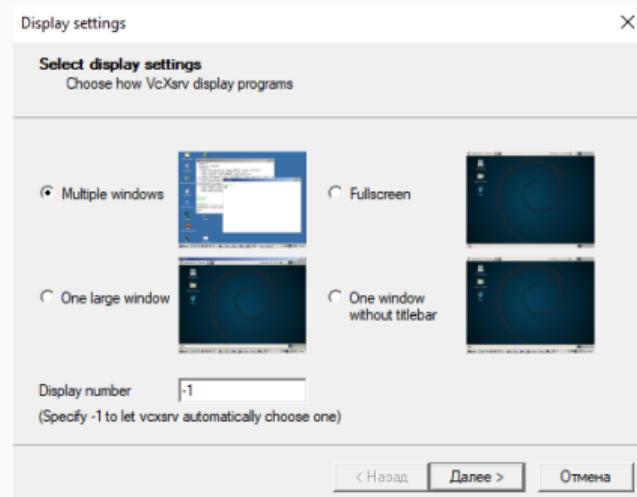


Рис. 10: Выбор опций

# Настройка стенда виртуальной машины Mininet

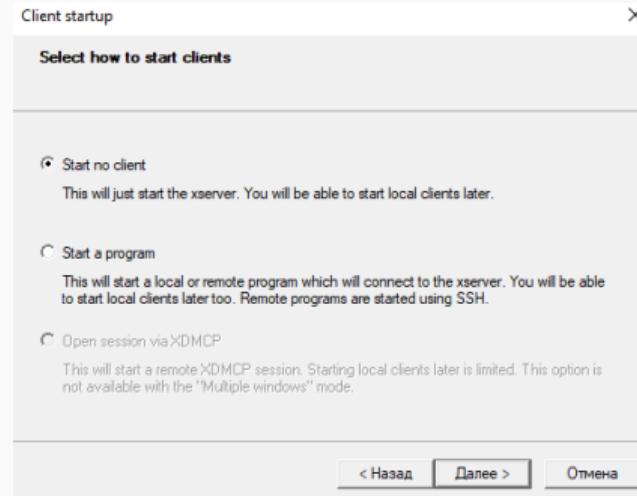


Рис. 11: Выбор опций

# Настройка стенда виртуальной машины Mininet

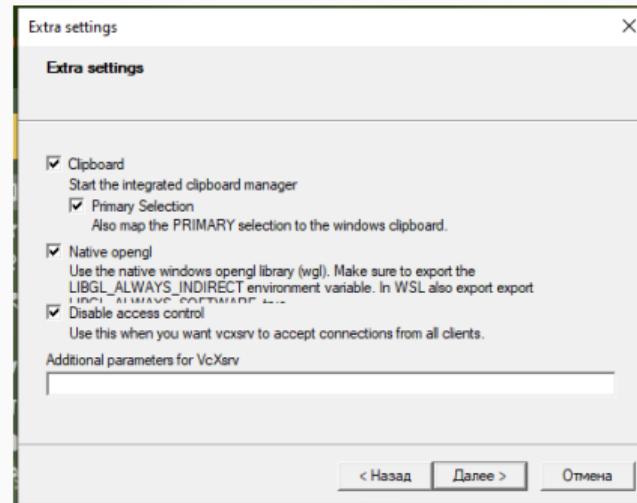


Рис. 12: Подключение в putty

# Настройка стенда виртуальной машины Mininet

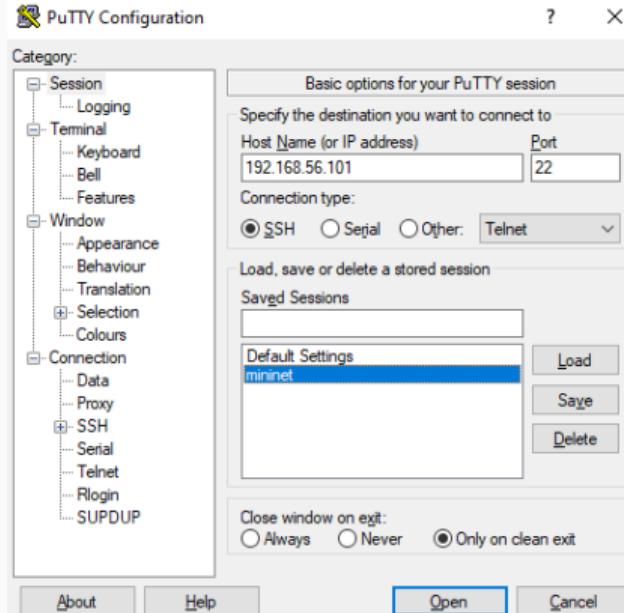


Рис. 13: Подключение в putty

# Настройка стенда виртуальной машины Mininet

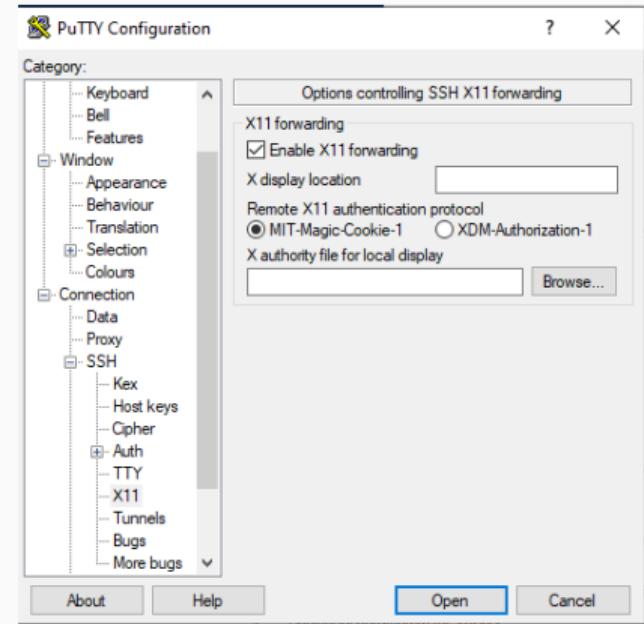
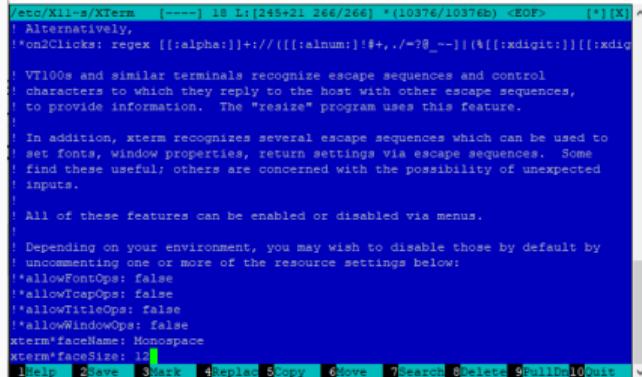


Рис. 14: Добавление опции перенаправления X11

# Настройка параметров XTerm



The screenshot shows an XTerm window with a blue background and white text. It displays configuration options for the XTerm application, specifically regarding escape sequences and font settings. The text is as follows:

```
/etc/X11-5/XTerm  [----] 18 L:[245+21 246/266] *(10376/10376b) <EOF> [*] [X] ^
Alternatively,
!*on2Clicks: regex [[[:alpha:]]+://([[[:alnum:]!#+,/.*?0_--])|(%[[[: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.

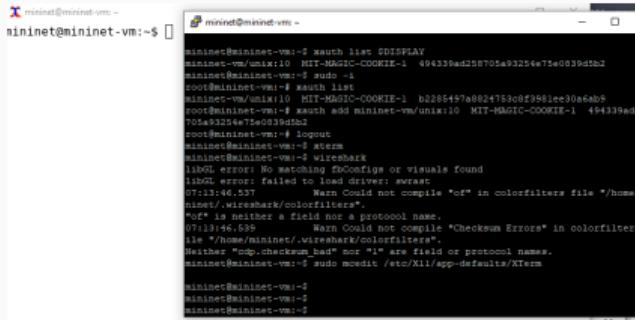
In addition, xterm recognizes several escape sequences which can be used to
set fonts, window properties, return settings via escape sequences. Some
find these useful; others are concerned with the possibility of unexpected
inputs.

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
*allowFcapOps: false
*allowTitleOps: false
*allowWindowOps: false
xterm*faceName: Monospace
xterm*faceSize: 12
1Help  2Save  3Mark  4Replace 5Copy  6Move  7Search 8Select 9Kill 10Quit
```

Рис. 15: Выбран системный моноширинный шрифт, кегль шрифта – 12 пунктов

# Настройка соединения X11 для суперпользователя



The screenshot shows a terminal window with two panes. The left pane displays a command-line session where the user copies a cookie value from one terminal window to another. The right pane shows the resulting xterm window where the user runs 'xterm'.

```
mininet@mininet-vm:~$ xauth list :0DISPLAY
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
mininet@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 b2205497a0024750d0f3991ec30a6ab9
root@mininet-vm:~# xterm
root@mininet-vm:~# logout
mininet@mininet-vm:~$ xterm
mininet@mininet-vm:~$ xterm
[1]  error: No matching EBCONFIG or visuals found
[2]  error: failed to open device: evdev
[3] 11:46.537  Warn Could not compile "of" in colorfilters file "/home/mininet/.wireshark/colorfilters".
Neither "tcp.checksum_bad" nor "!" are found or protocol names.
mininet@mininet-vm:~$ xterm
[4] 11:46.539  Warn Could not compile "Checksum Errors" in colorfilters
file "/home/mininet/.wireshark/colorfilters".
Neither "tcp.checksum_bad" nor "!" are found or protocol names.
mininet@mininet-vm:~$ xterm
[5] 11:46.540  Warn Could not compile "/etc/X11/app-defaults/Xterm"
mininet@mininet-vm:~$
```

Рис. 16: Копирование значения куки и запуск xterm

# Настройка соединения X11 для суперпользователя

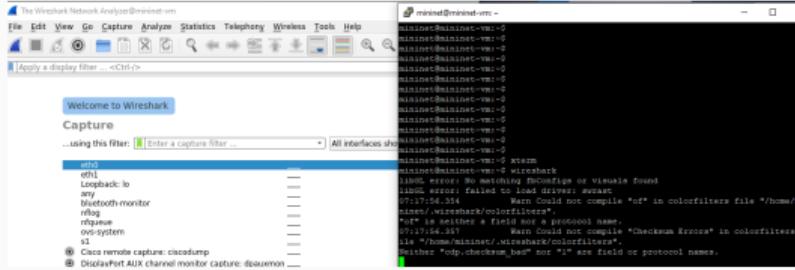


Рис. 17: Запуск Wireshark

# Настройка доступа к Интернет

```
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 26060 bytes 3138645 (3.1 MB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 44805 bytes 60236707 (60.2 MB)
                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 8544 bytes 59085104 (59.0 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8544 bytes 59085104 (59.0 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$
```

Рис. 18: Вывод команды ifconfig

# Настройка доступа к Интернет

```
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

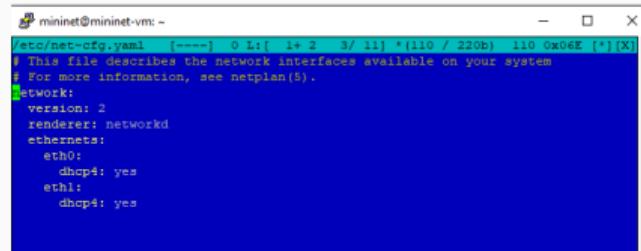
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
                ether 08:00:27:19:7d:07 txqueuelen 1000 (Ethernet)
                RX packets 392 bytes 38403 (38.4 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                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. 19: Активен адрес NAT 10.0.2.15

# Настройка доступа к Интернет



The screenshot shows a terminal window with the command `mininet@mininet-vm: ~`. The window displays the contents of the `/etc/netcfg.yaml` file. The file is a YAML configuration for network interfaces. It includes comments explaining the file's purpose and how to use netplan(5). The configuration section starts with `network:`, followed by `version: 2`, `renderer: networkd`, and `ethernets:`. Under `ethernets`, there are two entries: `eth0:` and `eth1:`, both with `dhcp4: yes`.

```
mininet@mininet-vm: ~
/etc/netcfg.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. 20: Внесение изменений в 01-netcfg.yaml

# Обновление версии Mininet

```
mininet@mininet-vn:~$ mv ~/mininet ~/mininet.org
mininet@mininet-vn:~$ cd ~
mininet@mininet-vn:~$ 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-vn:~$ cd ~/mininet
mininet@mininet-vn:~/mininet$ sudo make install
cc -Wall -Wextra \
-DVERSION="`PYTHONPATH=. python -B bin/mn --version 2>&1`" mnexec.c -o mnexec
```

Рис. 21: Установка новой версии Mininet

## Обновление версии Mininet

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

Рис. 22: Версия Mininet

# Основы работы в Mininet

```
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> [redacted]
```

Рис. 23: Запуск минимальной топологии и отображение списка команд интерфейса

# Основы работы в 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> 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. 24: Доступные узлы и отображение связи между устройствами

# Основы работы в Mininet

```
mininet> h2 ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      link layer address 56:84:7a brd ff:ff:ff:ff:ff:ff
      ether 56:84:7a:00:00:00 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overrun 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overrun 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      link layer address 00:00:00:00:00:00 brd 00:00:00:00:00:00
      ether 00:00:00:00:00:00 txqueuelen 1000 (Loopback)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overrun 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overrun 0 carrier 0 collisions 0

mininet> s1 ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      link layer address 08:00:23:74:04:35 brd ff:ff:ff:ff:ff:ff
      ether 08:00:23:74:04:35 txqueuelen 1000 (Ethernet)
      RX packets 26268 bytes 3162827 (3.1 MB)
      RX errors 0 dropped 0 overrun 0 frame 0
      TX packets 44959 bytes 40345239 (60.2 MB)
      TX errors 0 dropped 0 overrun 0 carrier 0 collisions 0

eth1: flags=41<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      link layer address 10:0:0:2:0:0 brd ff:ff:ff:ff:ff:ff
      ether 10:0:0:2:0:0 txqueuelen 1000 (Ethernet)
      RX packets 51015105 bytes 51015105 (59.1 MB)
      RX errors 493 dropped 0 overrun 0 frame 0
      TX packets 451 bytes 40862 (40.9 kB)
      TX errors 0 dropped 0 overrun 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      link layer address 00:00:00:00:00:00 brd 00:00:00:00:00:00
      ether 00:00:00:00:00:00 txqueuelen 1000 (Loopback)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overrun 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overrun 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      link layer address 92:a8:27:fe:29:55 txqueuelen 1000 (Ethernet)
      ether 92:a8:27:fe:29:55 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overrun 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overrun 0 carrier 0 collisions 0

s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      link layer address 3a:af:fd:71:bb:d1 txqueuelen 1000 (Ethernet)
      ether 3a:af:fd:71:bb:d1 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overrun 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overrun 0 carrier 0 collisions 0
```

Рис. 25: Конфигурация h2 и s1

# Основы работы в Mininet

```
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. 26: Пингование 10.0.0.2

# Основы работы в Mininet

Меню

File Edit Run Help

File Edit Run Help

```
root@mininet-vn: ~
 1: eth0: flags=4163<UP,BROADCAST,MULTICAST>  mtu 1500
        link layer 16 bytes
        brd 0xffffffff
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8755 bytes 56101840 (52.1 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
 2: eth1: flags=4163<UP,BROADCAST,MULTICAST>  mtu 1500
        link layer 16 bytes
        brd 0xffffffff
        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
 3: eth2: flags=4163<UP,BROADCAST,MULTICAST>  mtu 1500
        link layer 16 bytes
        brd 0xffffffff
        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=3.20 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=3.20 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=3.00 ms
...
11: 11:11:22:33:44:55:66:77:88:99:00
11:11:22:33:44:55:66:77:88:99:00:00
mininet> exit
logout
There are stopped jobs.
mininet@mininet-vn: ~$ cd /home/mininet/mininet/examples/mininedit.py
cd: cannot find directory /home/mininet/mininet/examples/mininedit.py
mininet>
```

Рис. 27: Запуск MiniEdit

# Основы работы в Mininet

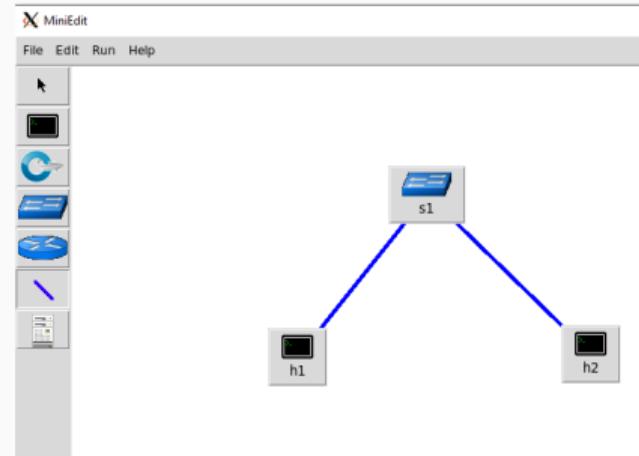


Рис. 28: Добавление двух хостов и одного коммутатора в рабочую область

# Основы работы в Mininet

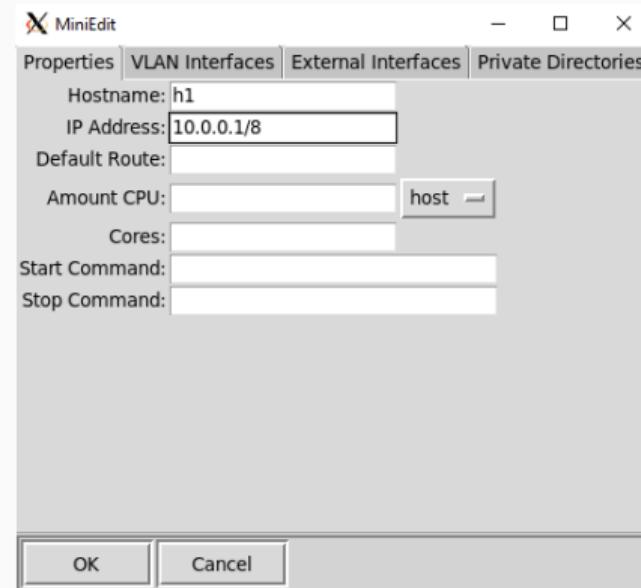


Рис. 29: Настройка IP-адреса на h1

# Основы работы в Mininet

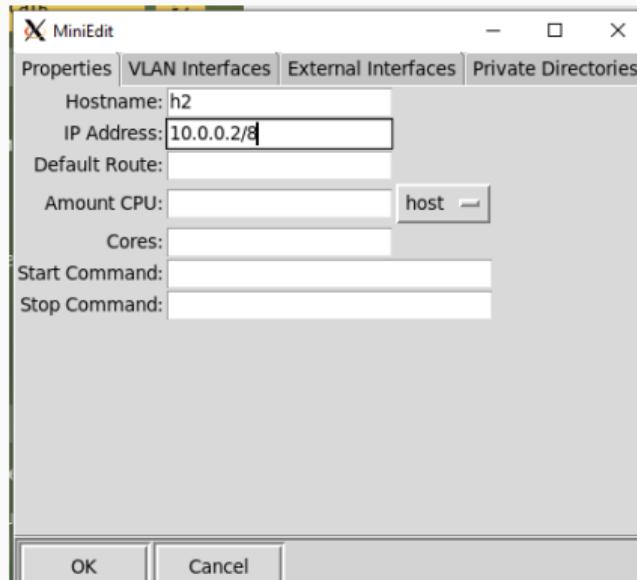


Рис. 30: Настройка IP-адреса на h2

# Основы работы в Mininet

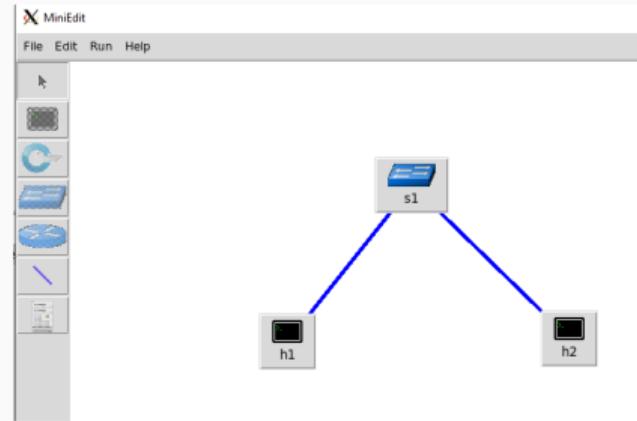


Рис. 31: Запуск эмуляции

# Основы работы в Mininet

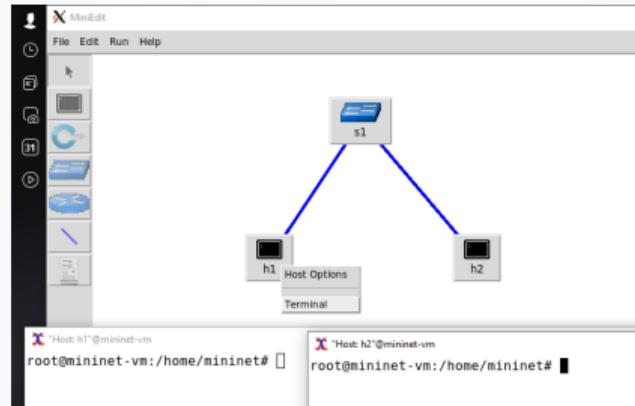


Рис. 32: Открытие терминала на h1 и h2

# Основы работы в Mininet

```
[root@mininet-vm ~]# mininet> ifconfig
h1-eth0: flags=413UNP, BRODCAST, MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
        brd 10.255.255.255 scope link
           link-layer brd 00:0c:29:00:00:01
              ether 00:0c:29:00:00:01 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=73UP,LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        brd 127.255.255.255 scope host
           link-layer brd 00:00:00:00:00:00
              ether 00:00:00:00:00:00 txqueuelen 1000
                 (Local Loopback)
             RX packets 27374 bytes 273876 (271.8 KB)
             RX errors 0 dropped 0 overruns 0 frame 0
             TX packets 27374 bytes 273876 (271.8 KB)
             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@mininet-vm ~]# mininet> ifconfig
h2-eth0: flags=413UNP, BRODCAST, MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
        brd 10.255.255.255 scope link
           link-layer brd 00:0c:29:00:00:02
              ether 00:0c:29:00:00:02 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=73UP,LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        brd 127.255.255.255 scope host
           link-layer brd 00:00:00:00:00:00
              ether 00:00:00:00:00:00 txqueuelen 1000
                 (Local Loopback)
             RX packets 269436 bytes 269436 (259.4 KB)
             RX errors 0 dropped 0 overruns 0 frame 0
             TX packets 269436 bytes 269436 (259.4 KB)
             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 33: Отображение, назначенных IP-адресов для h1 и h2

# Основы работы в Mininet

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

Рис. 34: Пингование 10.0.0.2

# Основы работы в Mininet

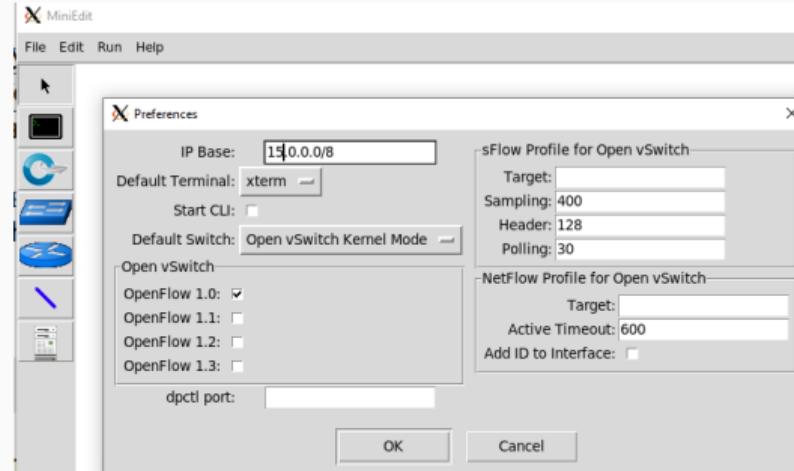


Рис. 35: Настройка автоматического назначения адресов на h1 и h2

# Основы работы в Mininet

```
[root@mininet-vm ~]# mininet -c
[root@mininet-vm ~]# ./home/mininet/firstcfg
h1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.1 brd 192.168.0.0 broadcast 192.168.0.0
              netmask 255.255.255.0
        ether 0a:0a:0a:00:00:01 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 brd 127.0.0.1 broadcast 127.0.0.1
              netmask 255.0.0.0
        ether 00:00:00:00:00:01 txqueuelen 1000  (Local Loopback)
      RX packets 1098 bytes 228128 (228.1 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 1098 bytes 228128 (228.1 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[root@mininet-vm ~]# ./home/mininet/firstcfg
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.2 brd 192.168.0.0 broadcast 192.168.0.0
              netmask 255.255.255.0
        ether a0:30:92:66: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 brd 127.0.0.1 broadcast 127.0.0.1
              netmask 255.0.0.0
        ether 00:00:00:00:00:01 txqueuelen 1000  (Local Loopback)
      RX packets 854 bytes 229988 (228.0 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 854 bytes 229988 (228.0 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[root@mininet-vm ~]#
```

Рис. 36: Успешное автоматическое назначение IP-адресов

# Основы работы в Mininet

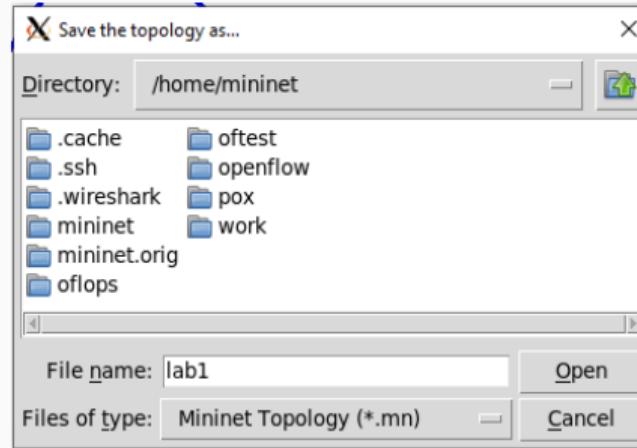


Рис. 37: Сохранение проекта

## Выводы

---

## Выводы

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