

Лабораторная работа №1

Введение в Mininet

Замбалова Д.В.

Российский университет дружбы народов, Москва, Россия

Докладчик

- Замбалова Дина Владимировна
- студентка
- Российский университет дружбы народов
- 1132226536@pfur.ru
- <https://github.com/dvzambalova>

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.

Выполнение лабораторной работы

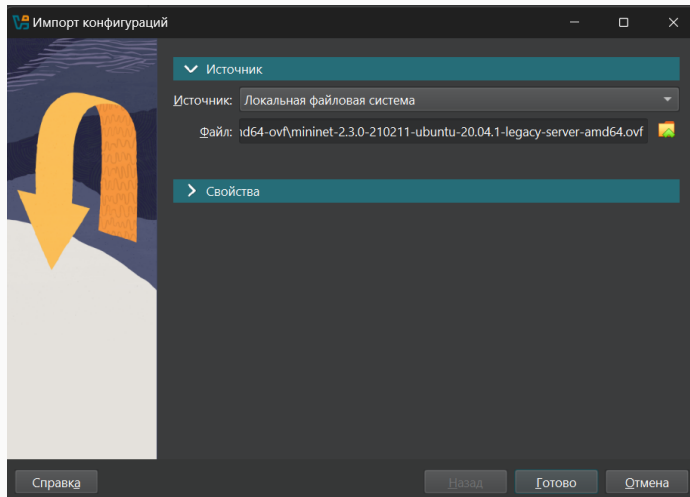


Рис. 1: Импорт конфигураций

Выполнение лабораторной работы

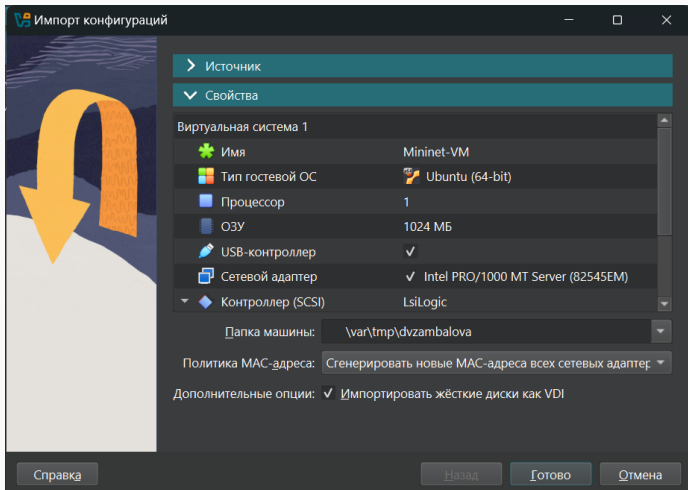


Рис. 2: Параметры импорта

Выполнение лабораторной работы

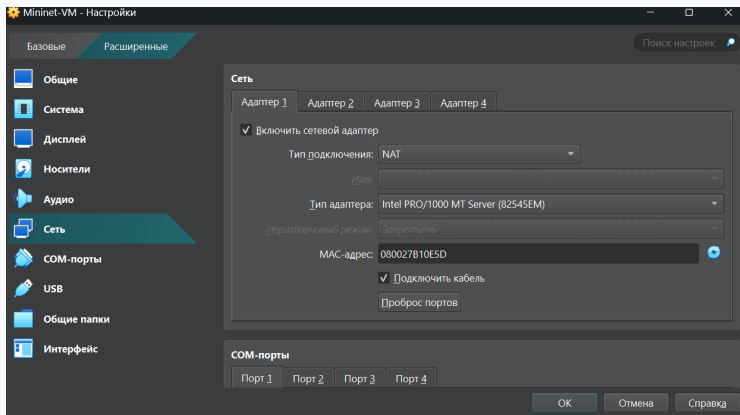


Рис. 3: Настройка сети

Выполнение лабораторной работы

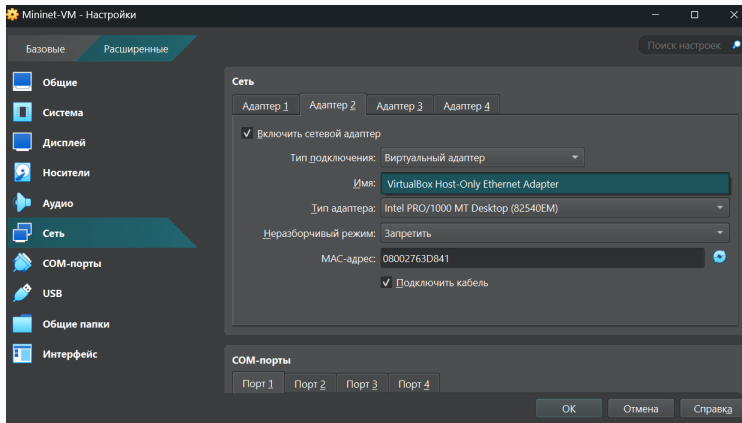
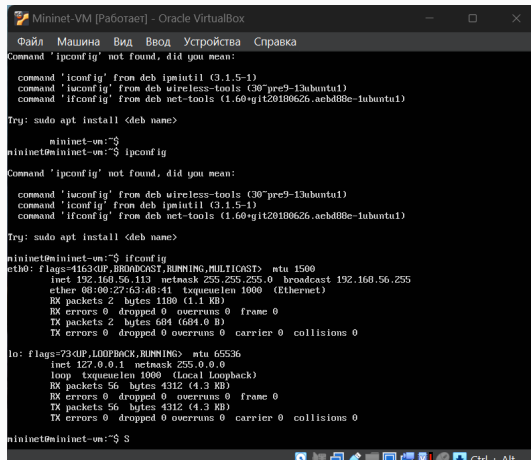


Рис. 4: Настройка сети

Выполнение лабораторной работы



```
Mininet-VM [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Command 'ipconfig' not found, did you mean:
  command 'iconfig' from deb ipmiutil (3.1.5-1)
  command 'isconfig' from deb wireless-tools (30~pre9-13ubuntu1)
  command 'ifconfig' from deb net-tools (1.60~git20180626.acbd88e-1ubuntu1)

Try: sudo apt install <deb name>

mininet-vm:~$
mininet@mininet-vm:~$ ipconfig

Command 'ipconfig' not found, did you mean:
  command 'isconfig' from deb wireless-tools (30~pre9-13ubuntu1)
  command 'iconfig' from deb ipmiutil (3.1.5-1)
  command 'ifconfig' from deb net-tools (1.60~git20180626.acbd88e-1ubuntu1)

Try: sudo apt install <deb name>

mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.113  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:63:d8:41  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 56  bytes 4312 (4.3 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 56  bytes 4312 (4.3 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$ S
```

Рис. 5: Запуск mininet

Выполнение лабораторной работы

```
zamba@LAPTOP-1M90JOT7 MSYS ~  
$ ssh-copy-id mininet@192.168.56.113  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zamba/.ssh/id_rsa.pub"  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys  
mininet@192.168.56.113's password:  
Permission denied, please try again.  
mininet@192.168.56.113's password:  
.  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'mininet@192.168.56.113'"  
and check to make sure that only the key(s) you wanted were added.  
  
zamba@LAPTOP-1M90JOT7 MSYS ~  
$ ssh -Y mininet@192.168.56.113  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings  
  
Last login: Sat Sep  6 06:20:29 2025 from 192.168.56.1  
mininet@mininet-vm:~$ |
```

Рис. 6: Подключение к mininet через SSH

Выполнение лабораторной работы

```
mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.113 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:63:d8:41 txqueuelen 1000 (Ethernet)
    RX packets 928 bytes 111199 (111.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 294 bytes 47359 (47.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    ether 08:00:27:b1:0e:5d 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 1840 bytes 141000 (141.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1840 bytes 141000 (141.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 7: Просмотр IP-адресов машины

Выполнение лабораторной работы

```
/etc/netplan/01-netcfg.yaml [-M--] 16 L:[ 1+ 9 10/ 11] *(219 / 220b)
# 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. 8: Файл /etc/netplan/01-netcfg.yaml

Выполнение лабораторной работы

```
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 | 2.91 MiB/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 .
^[[SProcessing /home/mininet/mininet
^[[SRequirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from mininet==2.3.1b4) (45.2.0)
Building wheels for collected packages: mininet
  Building wheel for mininet (setup.py) ... done
  Created wheel for mininet: filename=mininet-2.3.1b4-py3-none-any.whl size=160942 sha256=274a120e36ceb0b7e204b1a0bb2aa767f8728cb479499489fc5f4b760ae70d23
    Stored in directory: /tmp/pip-ephem-wheel-cache-j887q0b_/wheels/cd/7d/a7/aafef1b3eaff31efd6ba4e2eaf6c9690a717bdf739db6cfe8d45
Successfully built mininet
Installing collected packages: mininet
Successfully installed mininet-2.3.1b4
mininet@mininet-vm:~/mininet$ _
```

Рис. 9: Обновление Mininet

Выполнение лабораторной работы

```
node placement for cluster (experimental)?  
mininet@mininet-vm:~/mininet$ mn --version  
2.3.1b4  
mininet@mininet-vm:~/mininet$
```

Рис. 10: Номер установленной версии mininet

Выполнение лабораторной работы

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 4cbb57874d9b6a08ee805a5258a8c2f8
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 cfd80afdc298ef84233473ad48a3e0bd
root@mininet-vm:~#
root@mininet-vm:~# xauth add mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 4cbb57874d9b6a08ee805a5258a8c2f8
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 4cbb57874d9b6a08ee805a5258a8c2f8
root@mininet-vm:~# logout
mininet@mininet-vm:~$ sudo ~/mininet/mininet/examples/miniedit.py
topo=None
```

Рис. 11: Настройка соединения X11 для суперпользователя

Выполнение лабораторной работы

```
root@mininet-vn:~# 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
```

Рис. 12: Работа с Mininet с помощью командной строки

Выполнение лабораторной работы

```
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 ca:f4:8b:f1:5e:4d txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 13: Работа с Mininet с помощью командной строки

Выполнение лабораторной работы

```
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 8a:6f:fe:4d:39:79 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 192.168.56.113 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:63:d8:41 txqueuelen 1000 (Ethernet)
    RX packets 1933 bytes 191867 (191.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 988 bytes 136591 (136.5 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:b1:0e:5d txqueuelen 1000 (Ethernet)
    RX packets 4934 bytes 5917564 (5.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2863 bytes 184542 (184.5 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 1983 bytes 150926 (150.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1983 bytes 150926 (150.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 3a:85:4e:26:60:6b 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
```

Рис. 14: Работа с Mininet с помощью командной строки

Выполнение лабораторной работы

```
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=2.39 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.277 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.074 ms
^C64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.122 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9175ms
rtt min/avg/max/mdev = 0.068/0.332/2.394/0.689 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 281.211 seconds
```

Рис. 15: Проверка связности хостов

Выполнение лабораторной работы

```
root@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/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller ovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old x11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([_.,:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
```

Рис. 16: Очистка предыдущего экземпляра Mininet

Выполнение лабораторной работы

```
/etc/X11/app-defaults/XTerm [-M--] 15 L:[232*34 266/266] *(10373/10376b) 32 0x020 [*][X]
? Set a border for the menus to make them simpler to distinguish against the
? vt100 widget:
*SimpleMenu=borderWidth: 2

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

? Here is a pattern that is useful for double-clicking on a URL:
*charClass: 33:48,35:48,37-38:48,43-47:48,58:48,61:48,63-64:48,95:48,126:48
?
? Alternatively,
*on2Clicks: regex [[[:alpha:]]+://(?:[[:alnum:]]+|/|?@_|-|!%[:xdigit:]][:xdigit:]]+
?
? VT100s and similar terminals recognize escape sequences and control
? characters to which they reply to the host with other escape sequences,
? to provide information. The "resize" program uses this feature.
?
? In addition, xterm recognizes several escape sequences which can be used to
? set fonts, window properties, return settings via escape sequences. Some
? find these useful; others are concerned with the possibility of unexpected
? inputs.
?
? All of these features can be enabled or disabled via menus.
?
? Depending on your environment, you may wish to disable those by default by
? uncommenting one or more of the resource settings below:
*allowFontOps: false
*allowTcpOps: false
*allowTitleOps: false
*allowWindowOps: false
xterm=faceName: Monospace
xterm=faceSize: 12
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9FullDn 10Quit
```

Рис. 17: Настройка параметров XTerm

Выполнение лабораторной работы

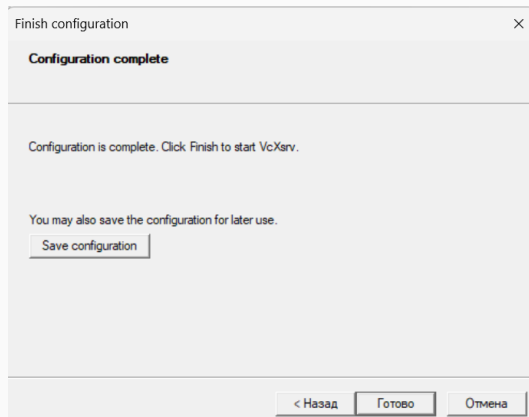


Рис. 18: Запуск и настройка Xserver

Выполнение лабораторной работы

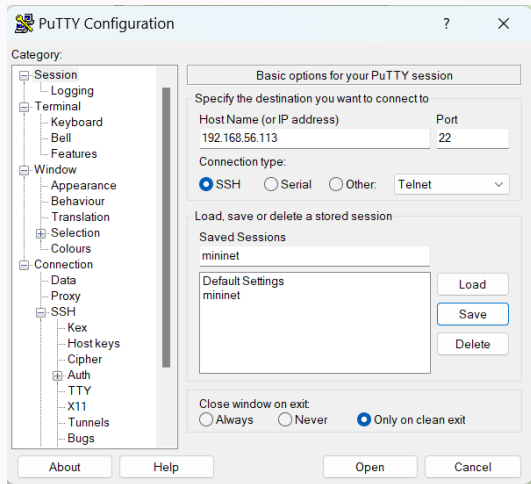


Рис. 19: Запуск putty и добавление опции перенаправления X11

Выполнение лабораторной работы

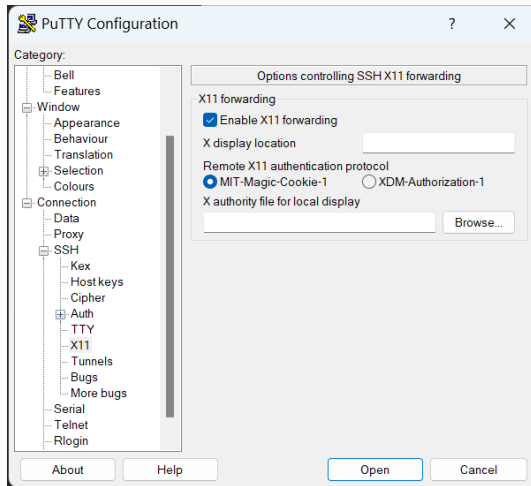


Рис. 20: Запуск putty и добавление опции перенаправления X11

Выполнение лабораторной работы

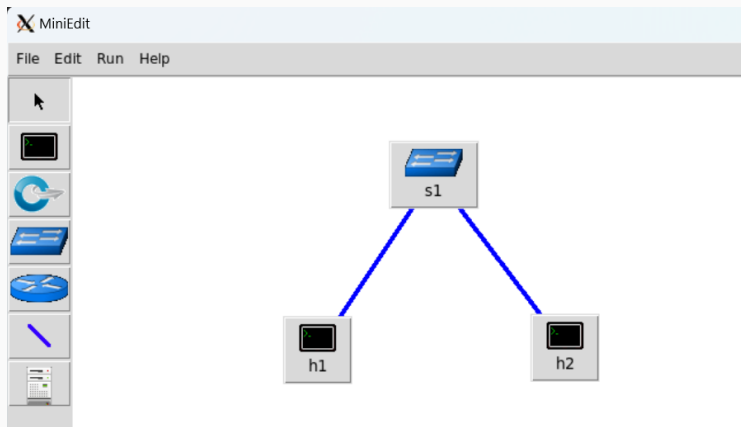
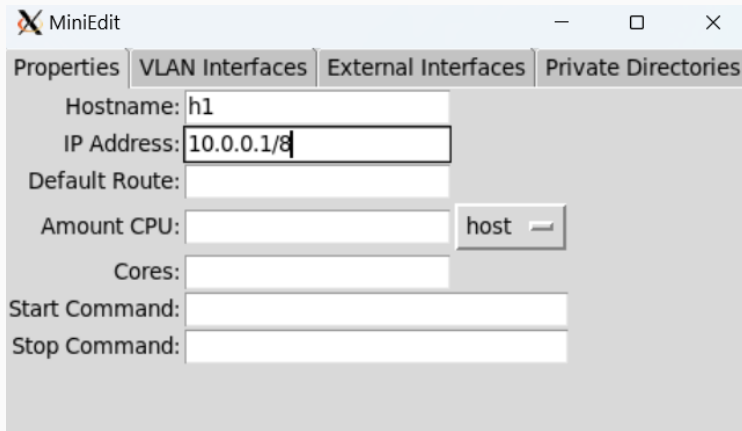


Рис. 21: Добавление двух хостов и одного коммутатора

Выполнение лабораторной работы



The image shows a window titled "MiniEdit" with four tabs: "Properties", "VLAN Interfaces", "External Interfaces", and "Private Directories". The "Properties" tab is active. It contains several configuration fields:

- Hostname: h1
- IP Address: 10.0.0.1/8 (this field is highlighted with a black border)
- Default Route: (empty field)
- Amount CPU: (empty field) and a dropdown menu showing "host"
- Cores: (empty field)
- Start Command: (empty field)
- Stop Command: (empty field)

Рис. 22: Настройка IP-адреса на хостах

Выполнение лабораторной работы

```
"Host: h1"@mininet-vm
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 4e:3f:5a:e7:d3:15 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 909 bytes 235384 (235.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 909 bytes 235384 (235.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.411 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.061 ms
```

Рис. 23: Проверка назначенных IP-адресов для h2 и проверка соединения между хостами

Выполнение лабораторной работы

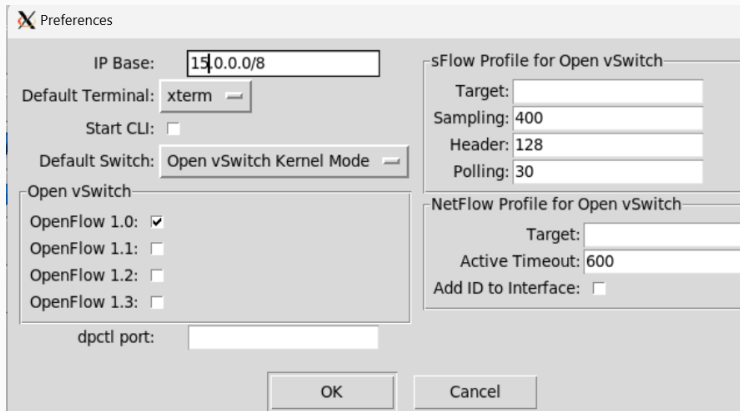
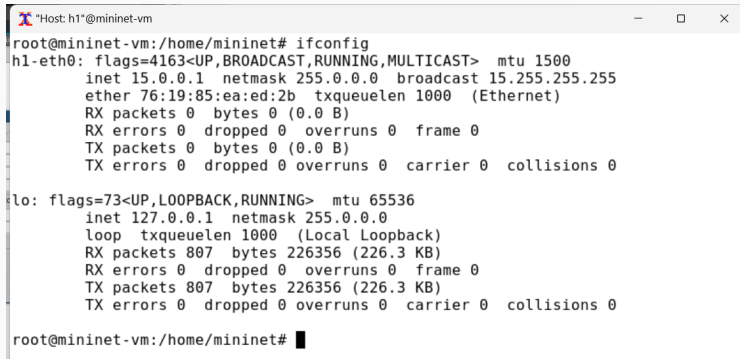


Рис. 24: Проверка автоматического назначения адресов

Выполнение лабораторной работы

A terminal window titled "Host: h1"@"mininet-vm" with standard window controls. It displays the output of the 'ifconfig' command for two interfaces: 'h1-eth0' and 'lo'. The 'h1-eth0' interface is configured with IP 15.0.0.1, netmask 255.0.0.0, and broadcast 15.255.255.255. The 'lo' interface is the loopback address 127.0.0.1. Statistics for both interfaces show zero errors and collisions.

```
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 76:19:85:ea:ed:2b 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 807 bytes 226356 (226.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 807 bytes 226356 (226.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 25: Отображение IP-адреса, назначенного хосту h1

Выполнение лабораторной работы

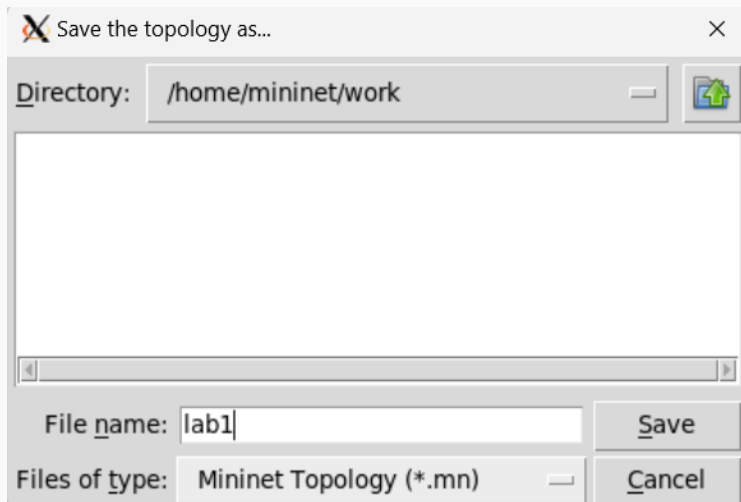


Рис. 26: Сохранение топологии

Выполнение лабораторной работы

```
mininet@mininet-vm:~$ ls -Al ~/work
total 4
-rw-r--r-- 1 root root 1655 Sep  7 05:32 lab1.mn
mininet@mininet-vm:~$ sudo chown -R mininet ~/work/
mininet@mininet-vm:~$
```

Рис. 27: Изменение прав доступа к файлам в каталоге проекта

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.

1. Mininet [Электронный ресурс]. Mininet Project Contributors. URL: <http://mini.net.org/> (дата обращения: 06.10.2025).

Спасибо за внимание!