Лабораторная работа № 1

Введение в Mininet

Шияпова Д.И.

20 мая 2025

Российский университет дружбы народов, Москва, Россия

Докладчик

- Шияпова Дарина Илдаровна
- Студентка
- Российский университет дружбы народов
- 1132226458@pfur.ru

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.

Mininet(**mininet?**) – это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины(ВМ). На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(ifconfig, ping) и даже получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик.

Mininet создает реалистичную виртуальную сеть, выполняя реальный код ядра, коммутатора и приложения на одной машине (VM, облачной или собственной) за считанные секунды с помощью одной команды `sudo mn`.

Выполнение лабораторной работы

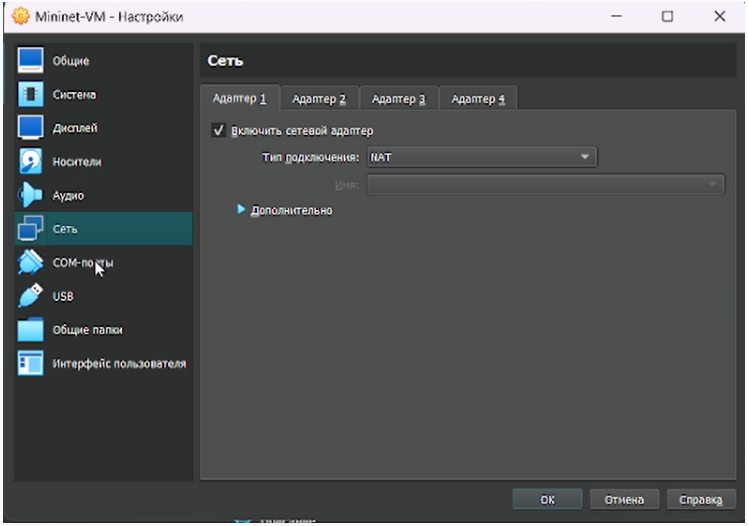


Рис. 1: Настройка сети

Выполнение лабораторной работы

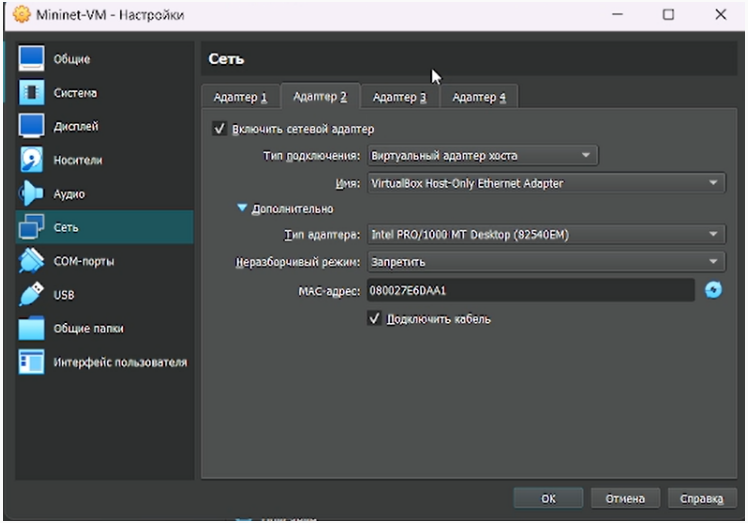
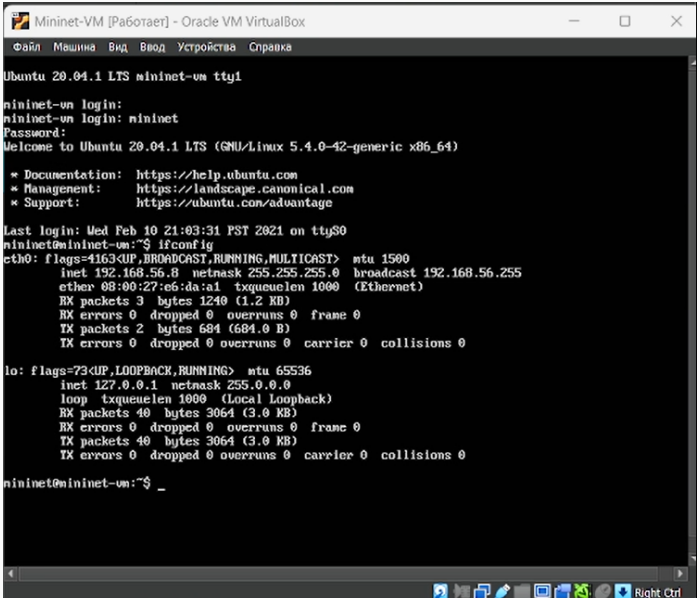


Рис. 2: Настройка сети

Выполнение лабораторной работы



Выполнение лабораторной работы

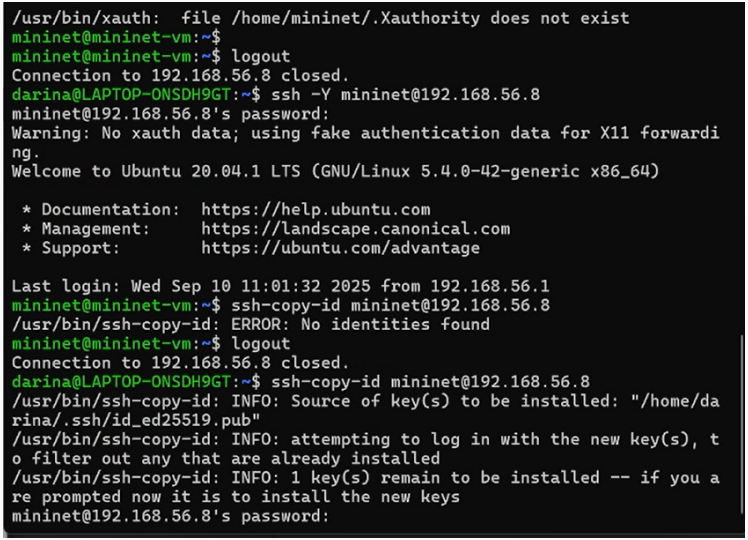
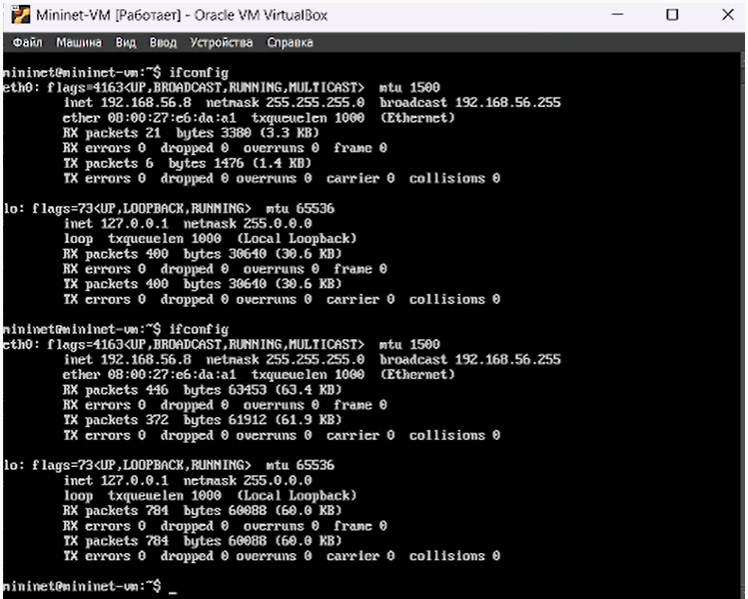


Рис. 4: Подключение к mininet через SSH

Выполнение лабораторной работы



Выполнение лабораторной работы

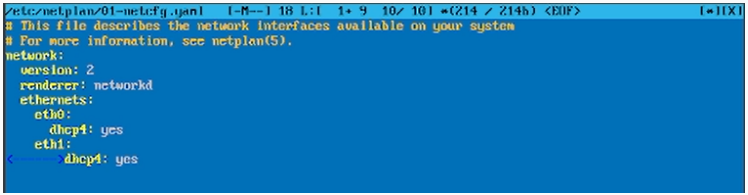
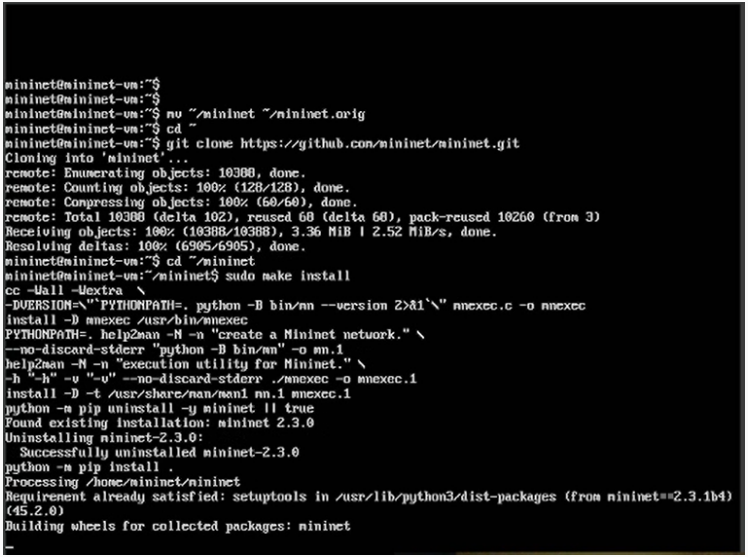


Рис. 6: Файл /etc/netplan/01-netcfg.yaml

Выполнение лабораторной работы



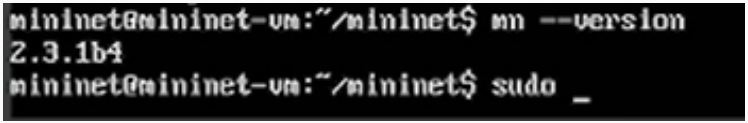


Рис. 8: Номер установленной версии mininet

Выполнение лабораторной работы

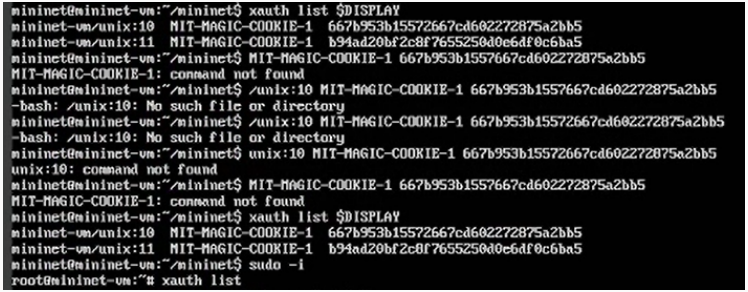


Рис. 9: Настройка соединения X11 для суперпользователя

Выполнение лабораторной работы

```
*** 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 b
  mininet> xterm h2
```

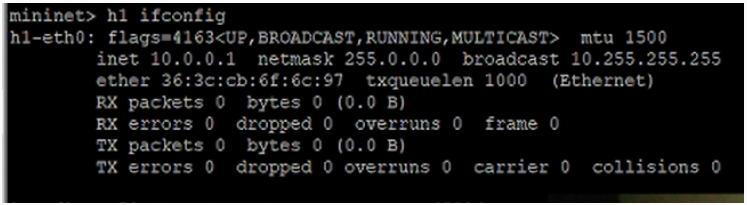


Рис. 11: Работа с Mininet с помощью командной строки

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 46:a0:db:37:7c:c6  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=13.4 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.200 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.0
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.0
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.0
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.0
^C
```

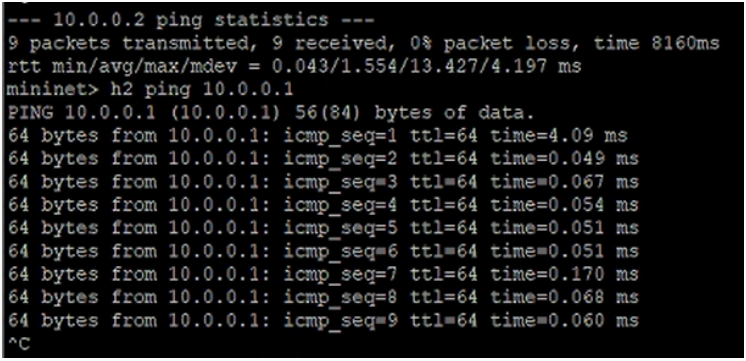


Рис. 13: Проверка связности хостов

Выполнение лабораторной работы

Рис. 14: sudo ~/mininet/mininet/examples/miniedit.py

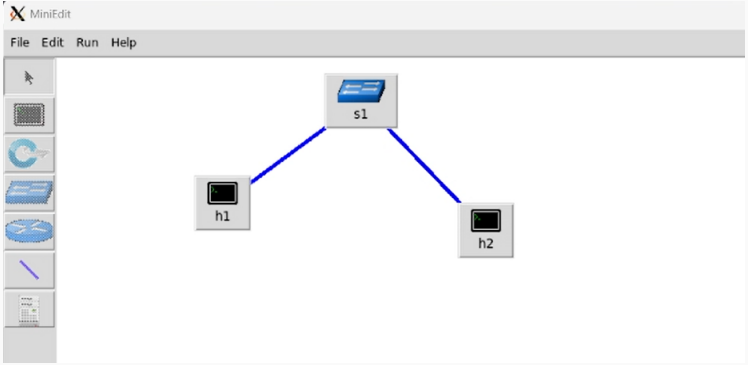


Рис. 15: Назначение IP-адресов

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.