Средство моделирования Mininet. Установка и пример моделирования.

Администрирование локальных сетей

Мишина А. А.

04 апреля 2025

Докладчик

- Мишина Анастасия Алексеевна
- НПИбд-02-22
- https://github.com/nasmi32

Введение

- Mininet – это виртуальная тестовая среда, предназначенная для разработки и тестирования сетевых инструментов и протоколов.
- Сети в Mininet создаются с помощью Python-скриптов.
- **Цель доклада:** разобраться с установкой Mininet на Windows, провести эксперимент.

Установка и настройка Mininet

Настройка образа Mininet

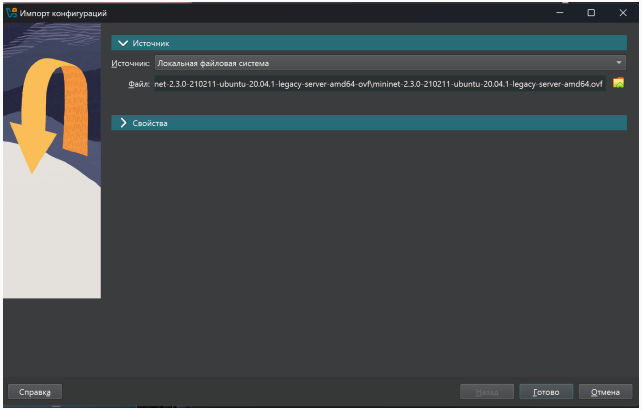

Рис. 1: Импорт образа ВМ

Настройка образа Mininet

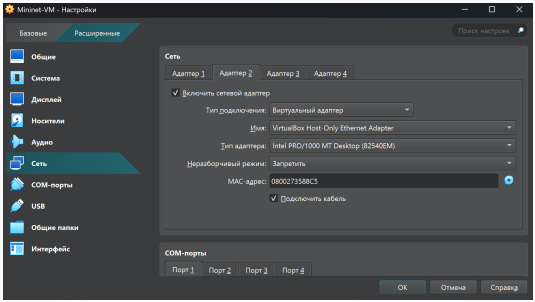

Рис. 2: Настройка сетевых параметров ВМ

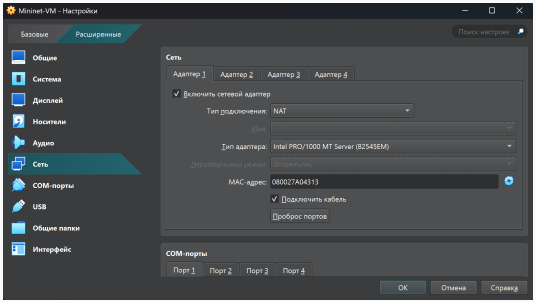

Рис. 3: Настройка сетевых параметров ВМ

Подключение к виртуальной машине

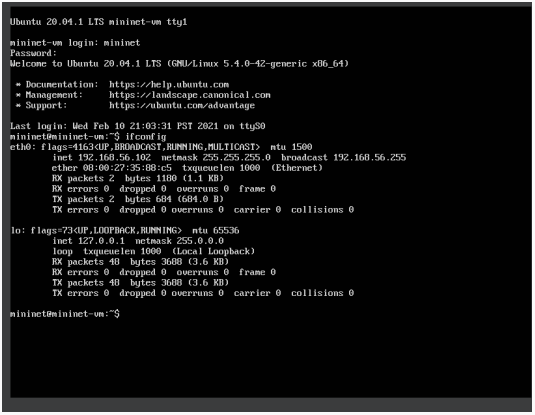

Рис. 4: Определение IP-адреса ВМ

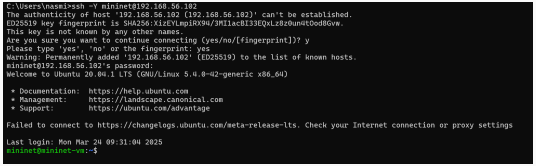

Рис. 5: Настройка SSH-подключения

```
choco install vcxsrv
```

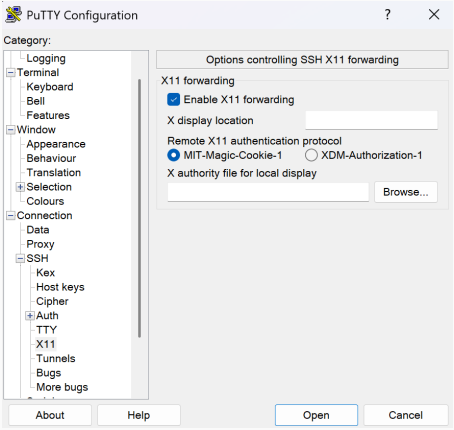

Рис. 6: Настройки Putty

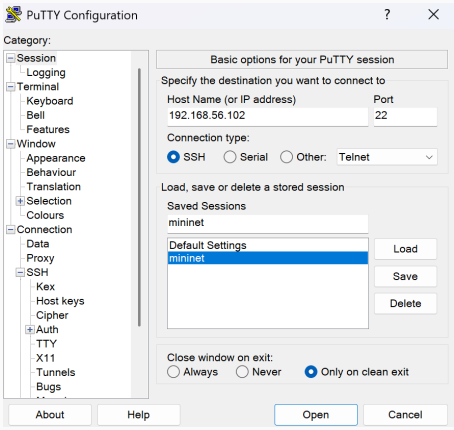

Рис. 7: Настройки Putty

Интернет

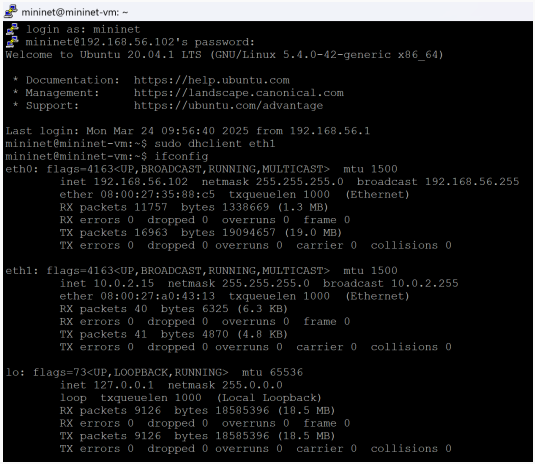

Рис. 8: Настройка доступа к интернету на ВМ

```
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>
```

Рис. 9: Запуск Mininet с минимальной топологией

```
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 a2:43:e6:38:1c:ec  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=10.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.393 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.137 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.141 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4076ms
rtt min/avg/max/mdev = 0.075/2.195/10.231/4.019 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 39.137 seconds
mininet@mininet-vm:~$
```

Рис. 10: Базовые команды Mininet

Практическая часть

Файл topology.py

```
net = Mininet(controller=Controller, waitConnected=True, link=TCLink)

info('*** Adding hosts\n')
h1 = net.addHost('h1', ip='10.0.1.2/24', defaultRoute='via 10.0.1.1')
h2 = net.addHost('h2', ip='10.0.2.2/24', defaultRoute='via 10.0.2.1')

info('*** Adding router\n')
router = net.addHost('router', ip='10.0.1.1/24')

info('*** Creating links\n')
net.addLink(h1, router, intfName2='router-eth1',
params2={'ip': '10.0.1.1/24'}, bw=100, delay='10ms')
net.addLink(h2, router, intfName2='router-eth2',
params2={'ip': '10.0.2.1/24'}, bw=100, delay='10ms')
```

Файл topology.py

```
info('*** Starting network\n')
net.start()

router.cmd('sysctl net.ipv4.ip_forward=1')

info('*** Starting iperf3 server on h2\n')
h2.cmd('iperf3 -s -D')
time.sleep(5)

info('*** h1 using iperf3\n')
h1.cmd(f'iperf3 -c {h2.IP()} -t 10 -l 1500 -J > iperf_result.json')

info('*** Stopping network\n')
net.stop()
```

Графики

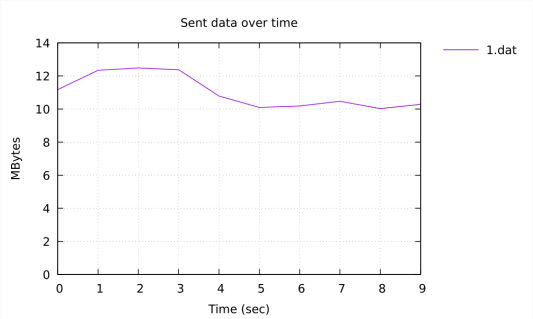

Рис. 11: График количества переданных байт

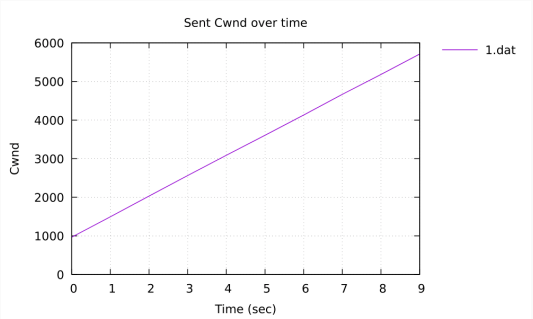

Рис. 12: График окна перегрузки

Графики

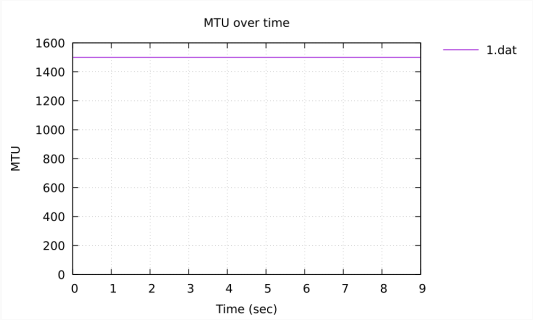

Рис. 13: График максимальной единицы передачи

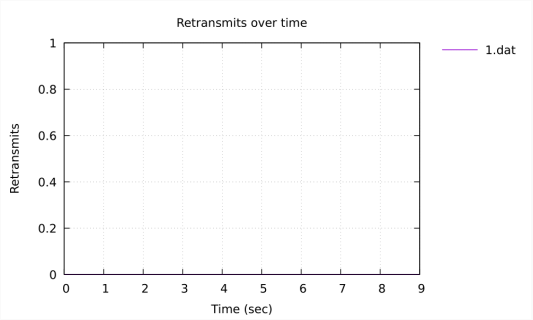

Рис. 14: График повторых передач

Графики

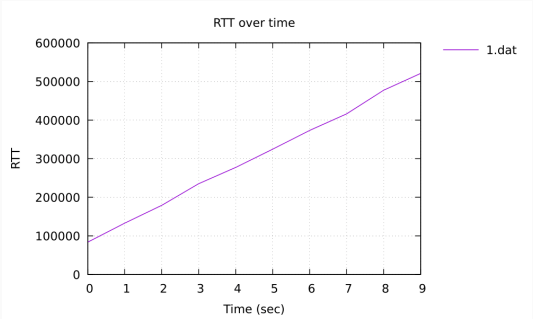

Рис. 15: График времени приема-передачи

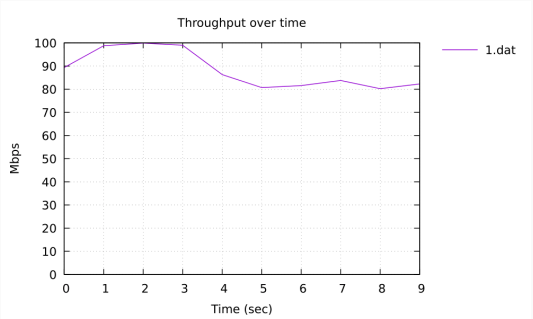

Рис. 16: График пропускной способности

Заключение

В ходе работы мы успешно развернули Mininet на платформе Windows и провели эксперимент по анализу пропускной способности сети.