

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина: Моделирование сетей передачи данных

Студент: Абрамян А. А.

Группа: НПИбд-01-20

МОСКВА

2023 г.

Цель работы

Целью данной 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.

Описание процесса выполнения работы

Постановка задачи

1. Настройка стенда виртуальной машины Mininet
2. Подключение к виртуальной машине
3. Основы работы в Mininet

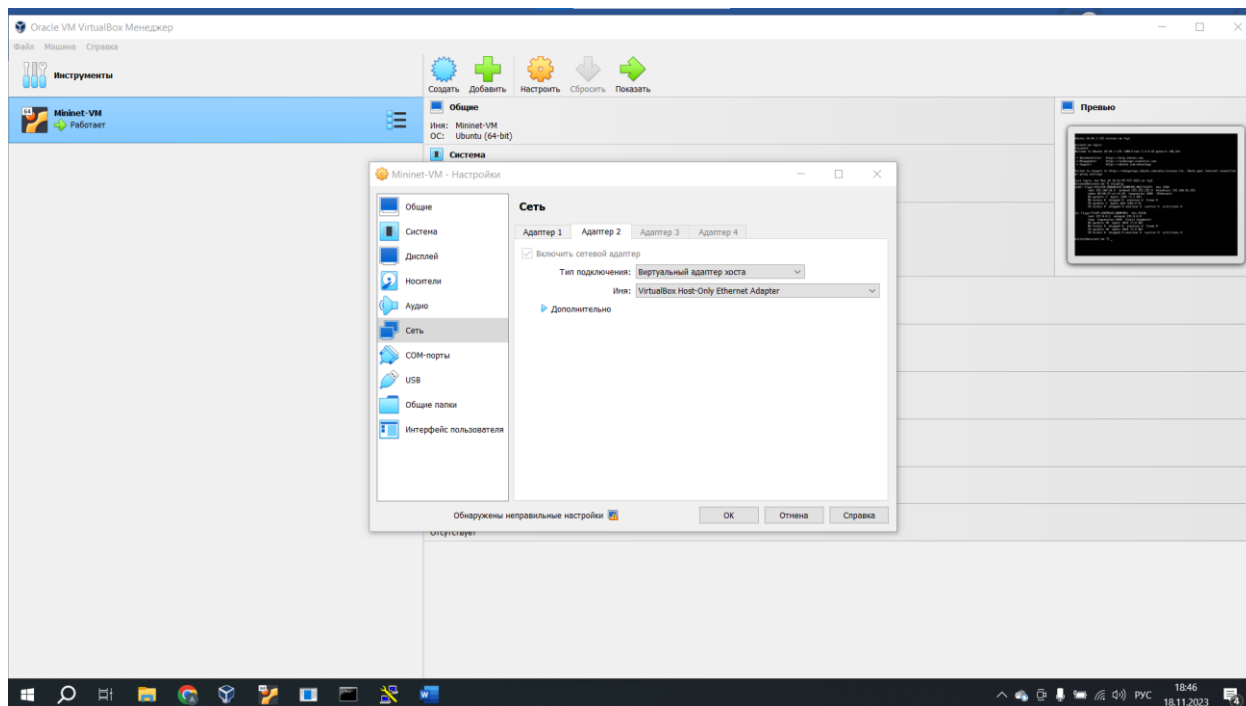
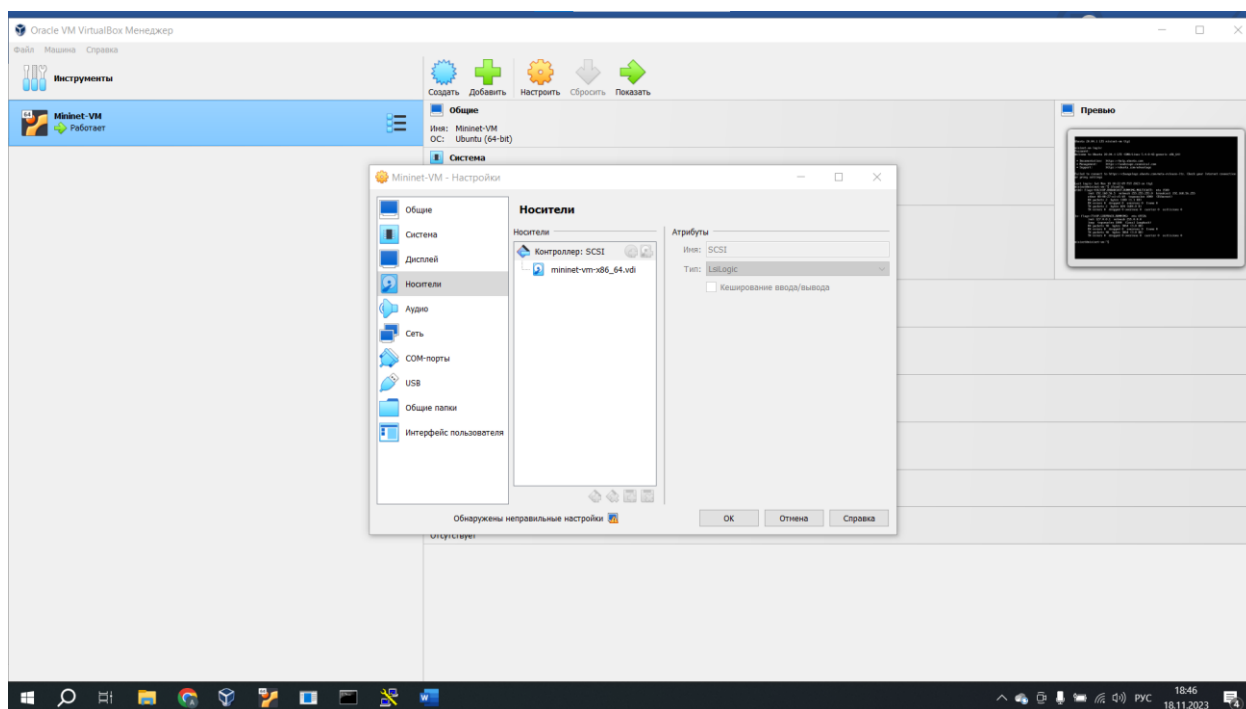
Порядок выполнения работы

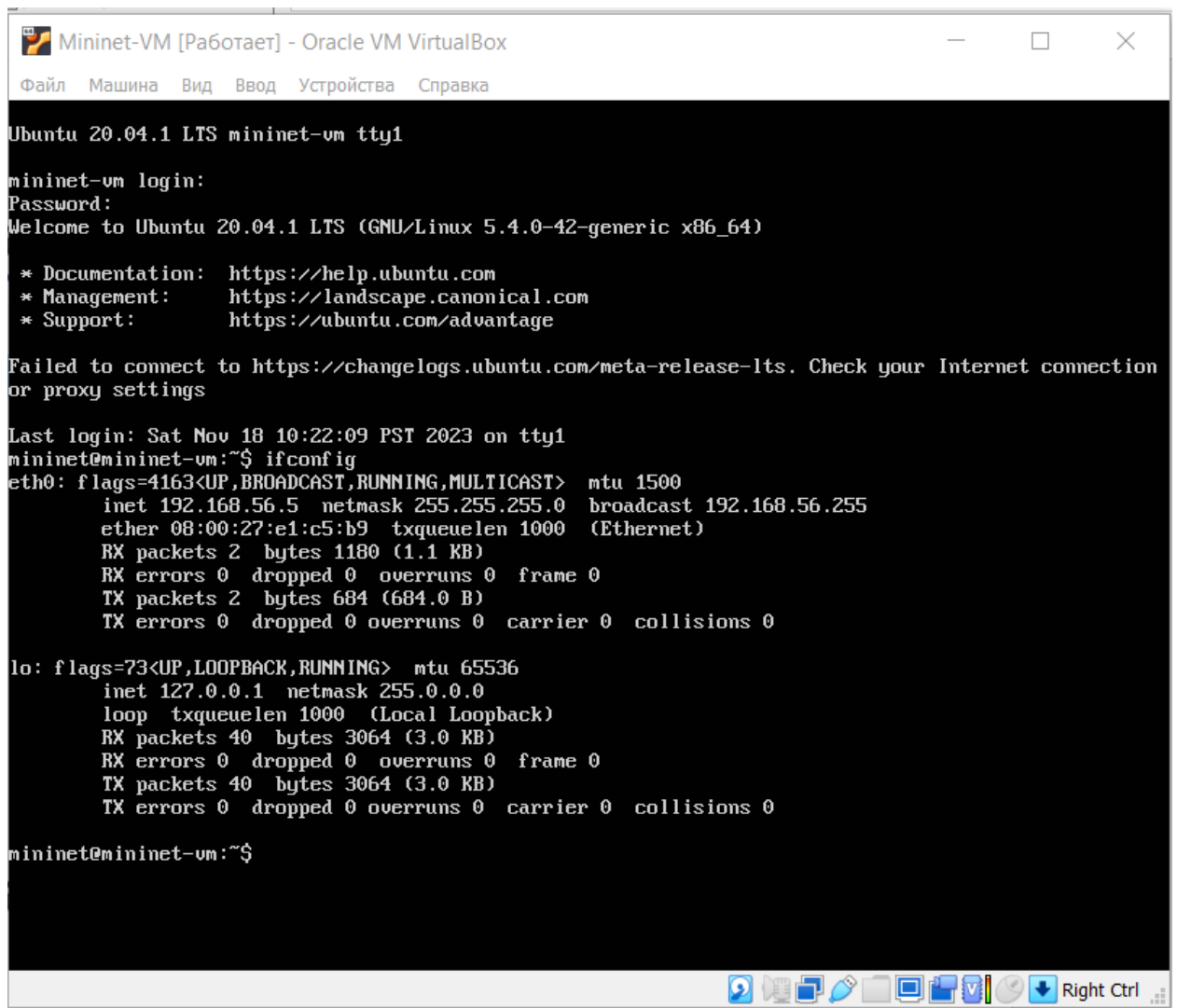
Настройка стенда виртуальной машины Mininet

Перейдите в репозиторий Mininet. Скачайте актуальный релиз ovf-образа виртуальной машины. – При необходимости переместите скачанный образ в каталог для работы, затем распакуйте его. – Запустите систему виртуализации и импортируйте файл .ovf. – Перейдите в настройки системы виртуализации и уточните параметры настройки виртуальной машины. В частности, для VirtualBox выберите импортированную виртуальную машину и перейдите в меню Машина Настроить . Перейдите к опции «Система». Если внизу этого окна есть сообщение об обнаружении неправильных настроек, то, следуя рекомендациям, внесите исправления. Например, может потребоваться увеличить видеопамять виртуальной машины и изменить тип графического контроллера на рекомендуемый. В настройках сети первый адаптер должен иметь подключение типа NAT. Для второго адаптера укажите тип подключения host-only network adapter (виртуальный адаптер хоста), который в дальнейшем вы будете использовать для входа в образ виртуальной машины. В этом режиме адаптер хоста использует специальное устройство vboxnet0, создает подсеть и назначает IP-адрес сетевой карте гостевой операционной системы. Если

данный режим не получается выбрать в настройках, то воспользуйтесь менеджером сетей хоста в VirtualBox для создания сети и настройки адаптера.

– Запустите виртуальную машину с Mininet.





```
Mininet-VM [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

Ubuntu 20.04.1 LTS mininet-vm tty1

mininet-vm login:
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: Sat Nov 18 10:22:09 PST 2023 on tty1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.5  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:e1:c5:b9  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

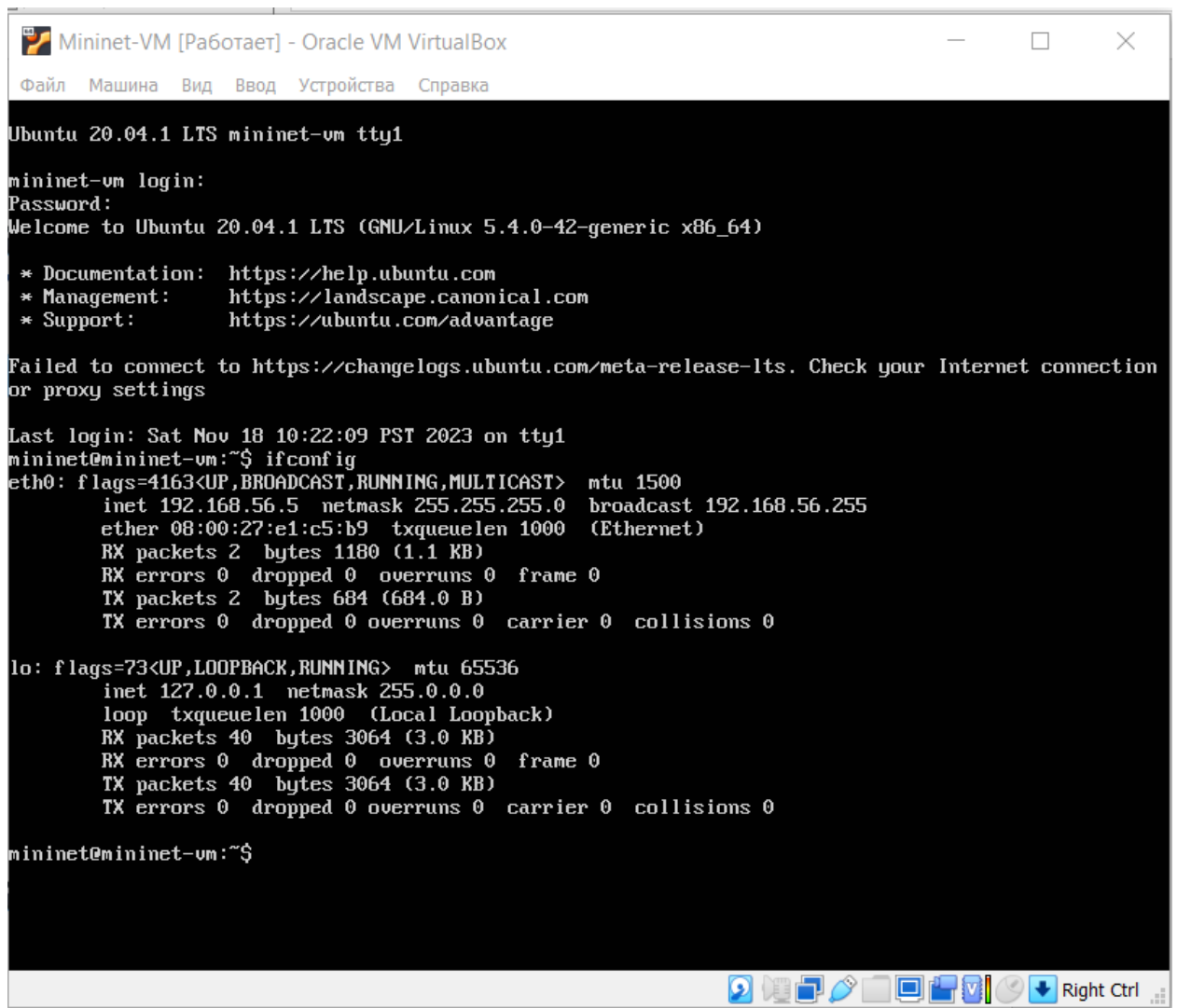
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop txqueuelen 1000  (Local Loopback)
    RX packets 40  bytes 3064 (3.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 40  bytes 3064 (3.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$
```

Подключение к виртуальной машине

Залогиньтесь в виртуальной машине: – login: mininet – password: mininet –

Посмотрите адрес машины: ifconfig



```
Mininet-VM [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

Ubuntu 20.04.1 LTS mininet-vm tty1

mininet-vm login:
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: Sat Nov 18 10:22:09 PST 2023 on tty1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.5  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:e1:c5:b9  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 40  bytes 3064 (3.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 40  bytes 3064 (3.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$
```

Подключитесь к виртуальной машине (из терминала хостовой машины):

```
(684 0 R)
mininet@mininet-vm: ~
login as: mininet
mininet@192.168.56.5'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

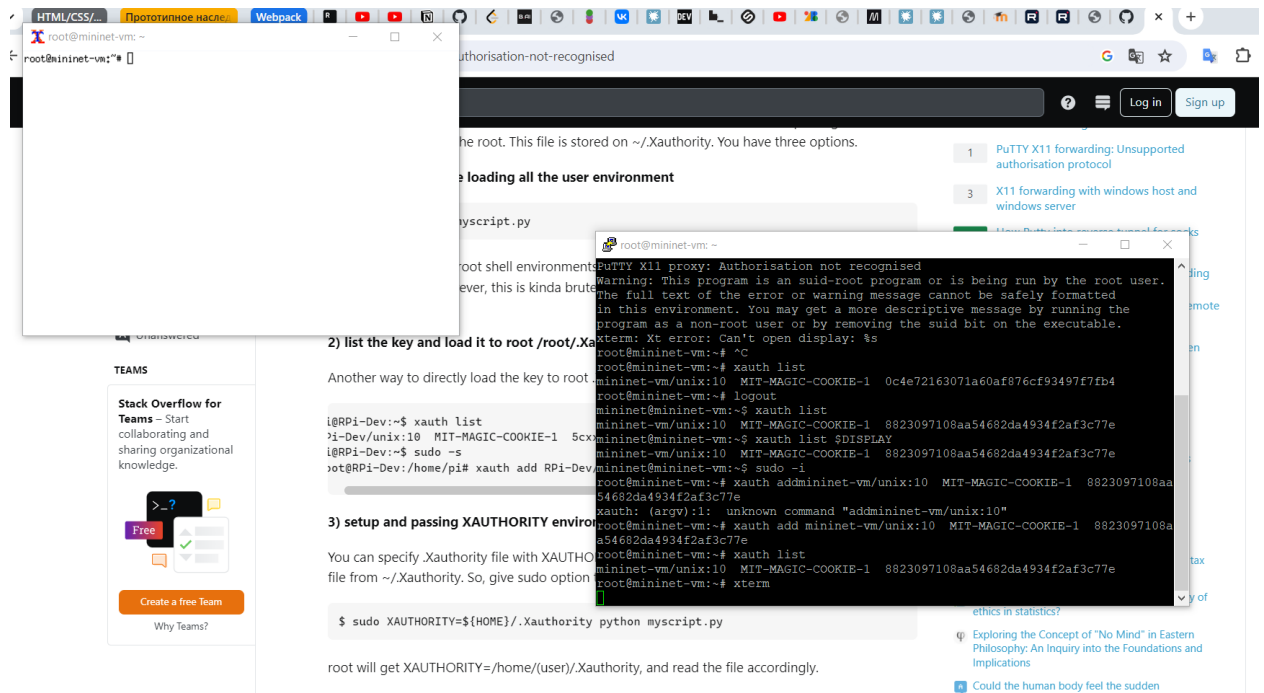
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Sat Nov 18 10:29:09 2023
mininet@mininet-vm:~$
```

Основы работы в Mininet

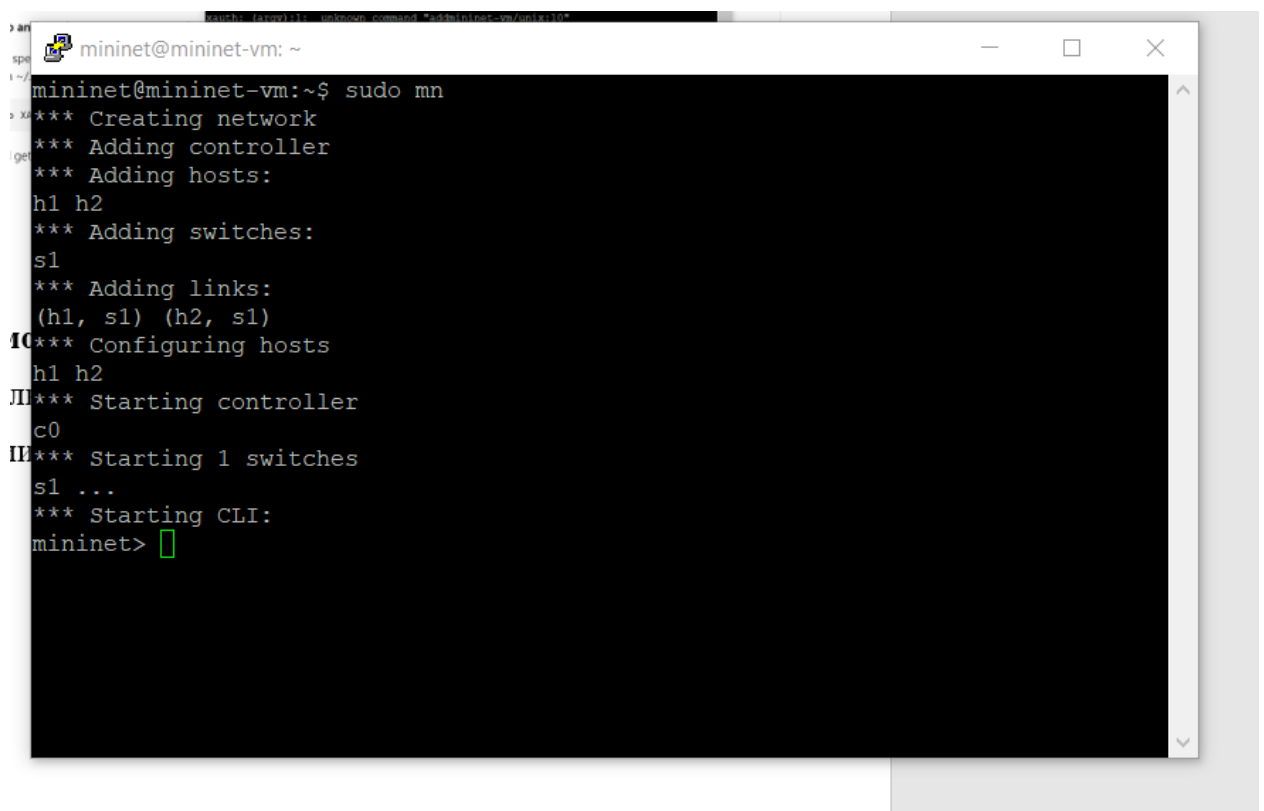
Настройка соединения X11 для суперпользователя

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. Скопируйте значение куки (MIT magic cookie)¹ пользователя mininet в файл для пользователя root:

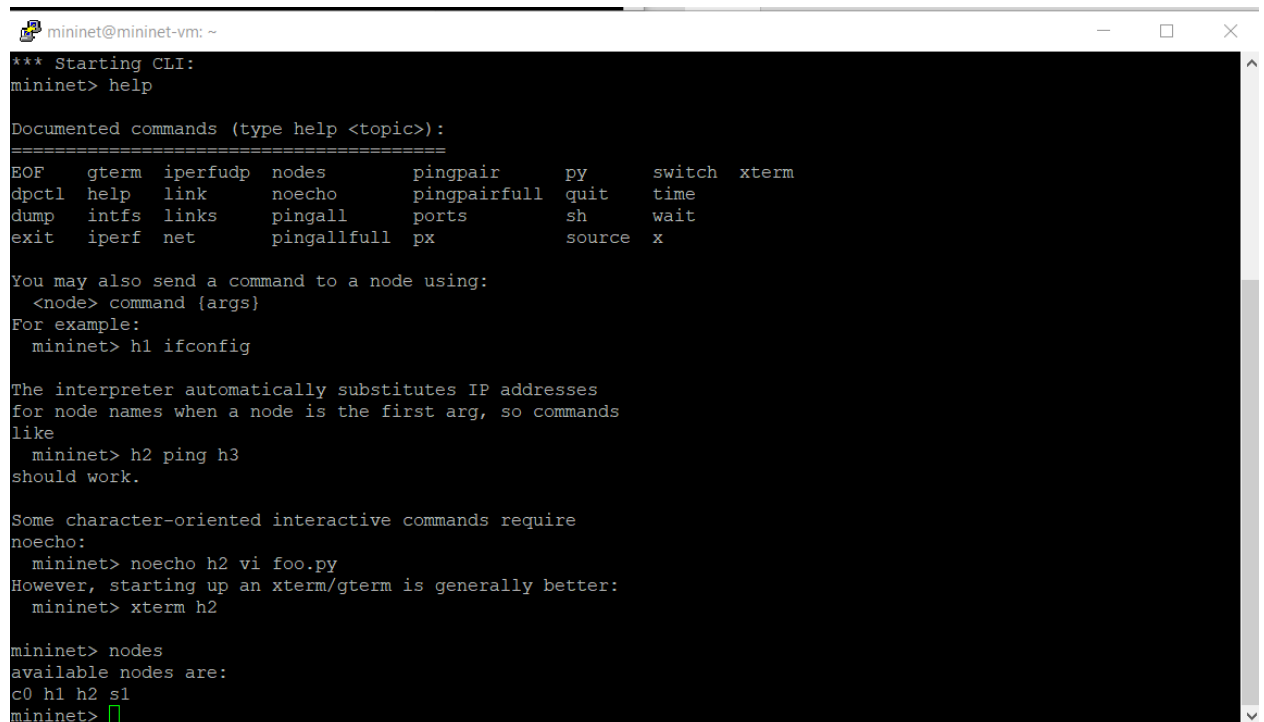


Работа с Mininet с помощью командной строки

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` – Для отображения доступных узлов введите: `nodes`



```
mininet@mininet-vm: ~
*** 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>
```

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`



```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet>
```

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.

Mininet позволяет выполнять команды на конкретном устройстве. Чтобы выполнить команду для определенного узла, необходимо сначала указать устройство, а затем команду, например:

h1 ifconfig

```
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 a6:27:a4:15:0a:e2 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>
```

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`.

```
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 4a:19:8f:5f:78:e1 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>
```

Эта запись выполняет 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`.

```

mininet> s1 ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.5 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:e1:c5:b9 txqueuelen 1000 (Ethernet)
    RX packets 10457 bytes 986764 (986.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15606 bytes 19502860 (19.5 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 5789 bytes 18989060 (18.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5789 bytes 18989060 (18.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 5a:82:89:22:83:cd 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 a2:3e:ec:7d:f2:2c 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>

```

Эта запись выполняет команду `ifconfig` на коммутаторе `s1` и показывает Коммутатор `s1`: – имеет петлевой интерфейс `lo`. – подключается к `h1-eth0` через интерфейс `s1-eth1`. – подключается к `h2-eth0` через интерфейс `s1-eth2`.

Проверка связности.

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, вы можете использовать команду `ping`. Команда `ping` работает, отправляя сообщения эхо-запроса протокола управляющих сообщений Интернета (ICMP) на удалённый компьютер и ожидая ответа. Например, команда `h1 ping 10.0.0.2` проверяет соединение между хостами `h1` и `h2`. Для остановки теста нажмите `Ctrl + c`.

```

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=1.76 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.149 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.030 ms
^C
--- 10.0.0.2 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12248ms
rtt min/avg/max/mdev = 0.030/0.177/1.758/0.457 ms
mininet>

```

Остановка эмуляции.

Для завершения работы режима эмуляции mininet используйте команду `exit`. Заметим, что команда `sudo mn -c` часто используется в терминале для очистки предыдущего экземпляра Mininet (например, после сбоя).

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 505.533 seconds
mininet@mininet-vm:~$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openflowd ovs-controllerovs-testcontroller udpbwtest
nexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openflowd ovs-controllerovs-testcontroller udpbwtest
t 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 | grep -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 | grep -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.
mininet@mininet-vm:~$
```

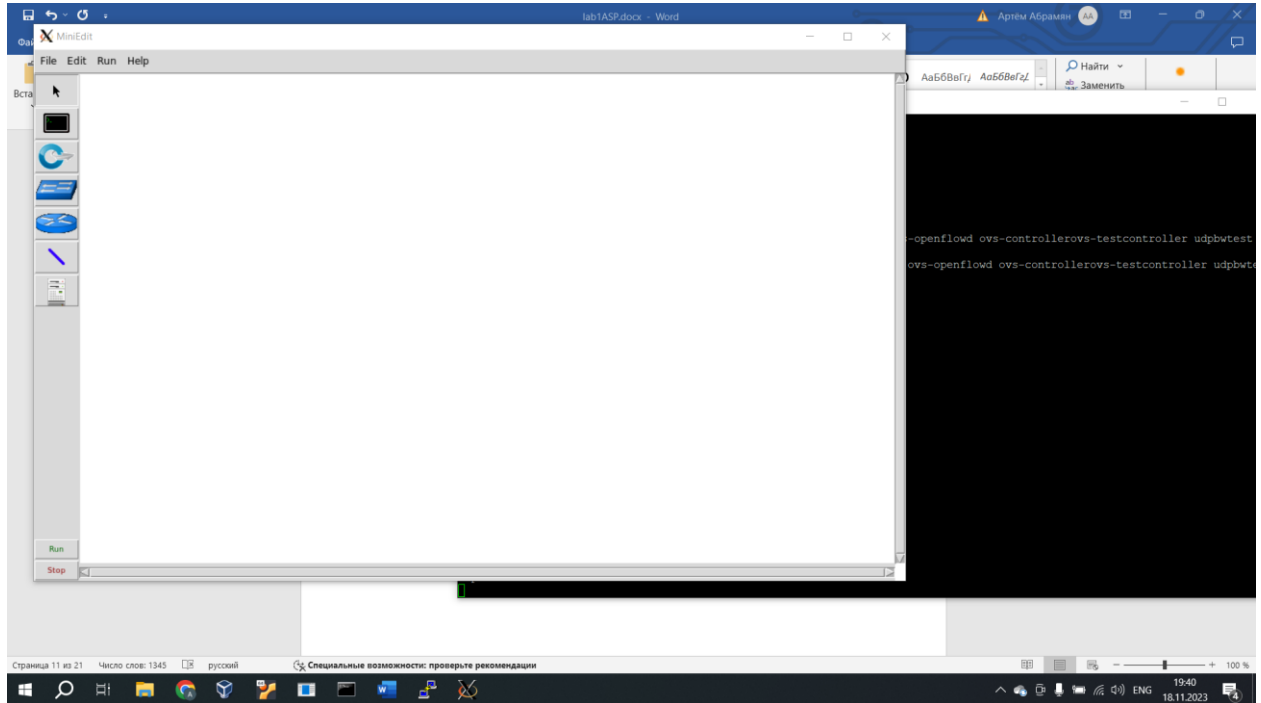
Построение и эмуляция сети в Mininet с использованием графического интерфейса

Построение топологии сети.

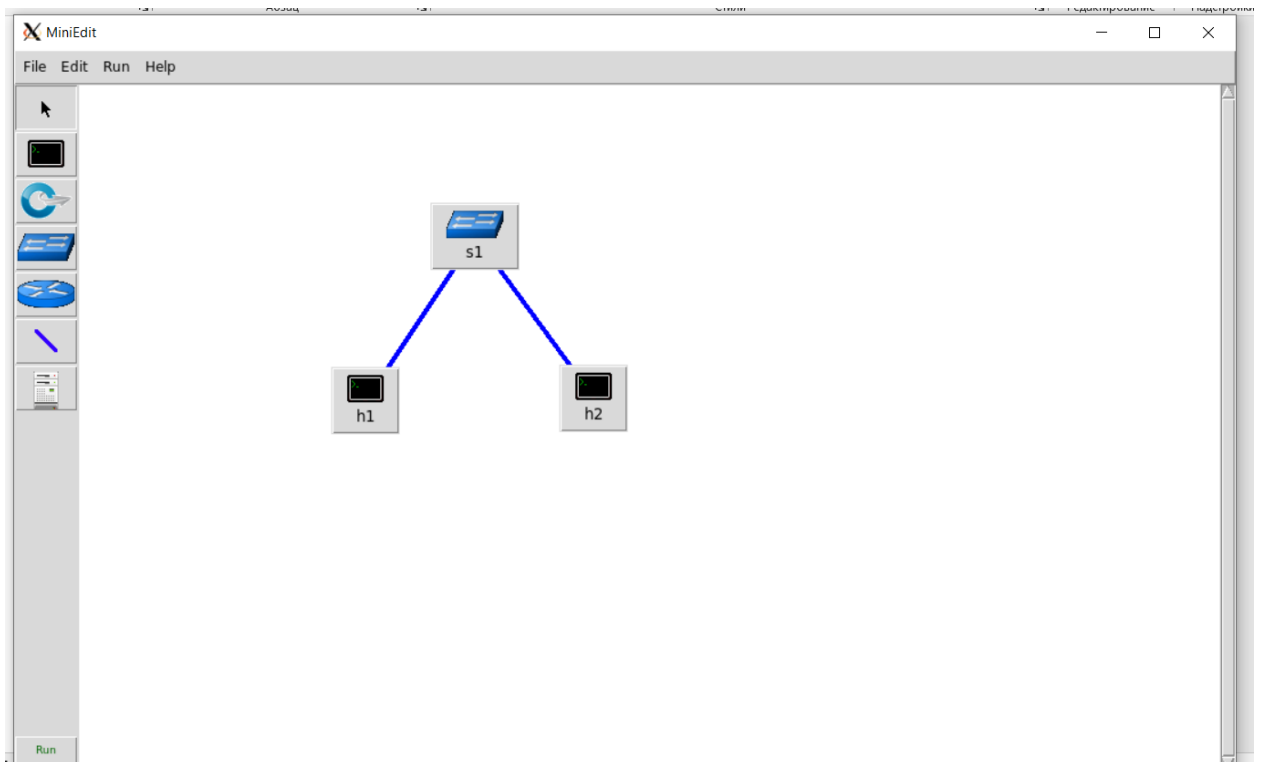
В терминале виртуальной машины mininet запустите MiniEdit: `sudo ~/mininet/mininet/examples/miniedit.py`. Основные кнопки: – Select: позволяет выбирать/перемещать устройства. Нажатие Del на клавиатуре после выбора устройства удаляет его из топологии. – Host: позволяет добавить новый хост в топологию. После нажатия этой кнопки щелкните в любом месте пустого холста, чтобы вставить новый хост. – Switch: позволяет добавить в топологию новый коммутатор. После нажатия этой кнопки щёлкните в любом месте пустого холста, чтобы вставить переключатель.

– Link: соединяет устройства в топологии. После нажатия этой кнопки

щелкните устройство и перетащите его на второе устройство, с которым необходимо установить связь. — Run: запускает эмуляцию. После проектирования и настройки топологии нажмите кнопку запуска. — Stop: останавливает эмуляцию.



Добавьте два хоста и один коммутатор, соедините хосты с коммутатором. — Настройте IP-адреса на хостах h1 и h2. Для этого удерживая правую кнопку мыши на устройстве выберите свойства. Для хоста h1 укажите IP-адрес 10.0.0.1/8, а для хоста h2 — 10.0.0.2/8.



12 10.0.0.2/8.

MiniEdit

Properties | **VLAN Interfaces** | External Interfaces | Private Directories

Hostname:

IP Address:

Default Route:

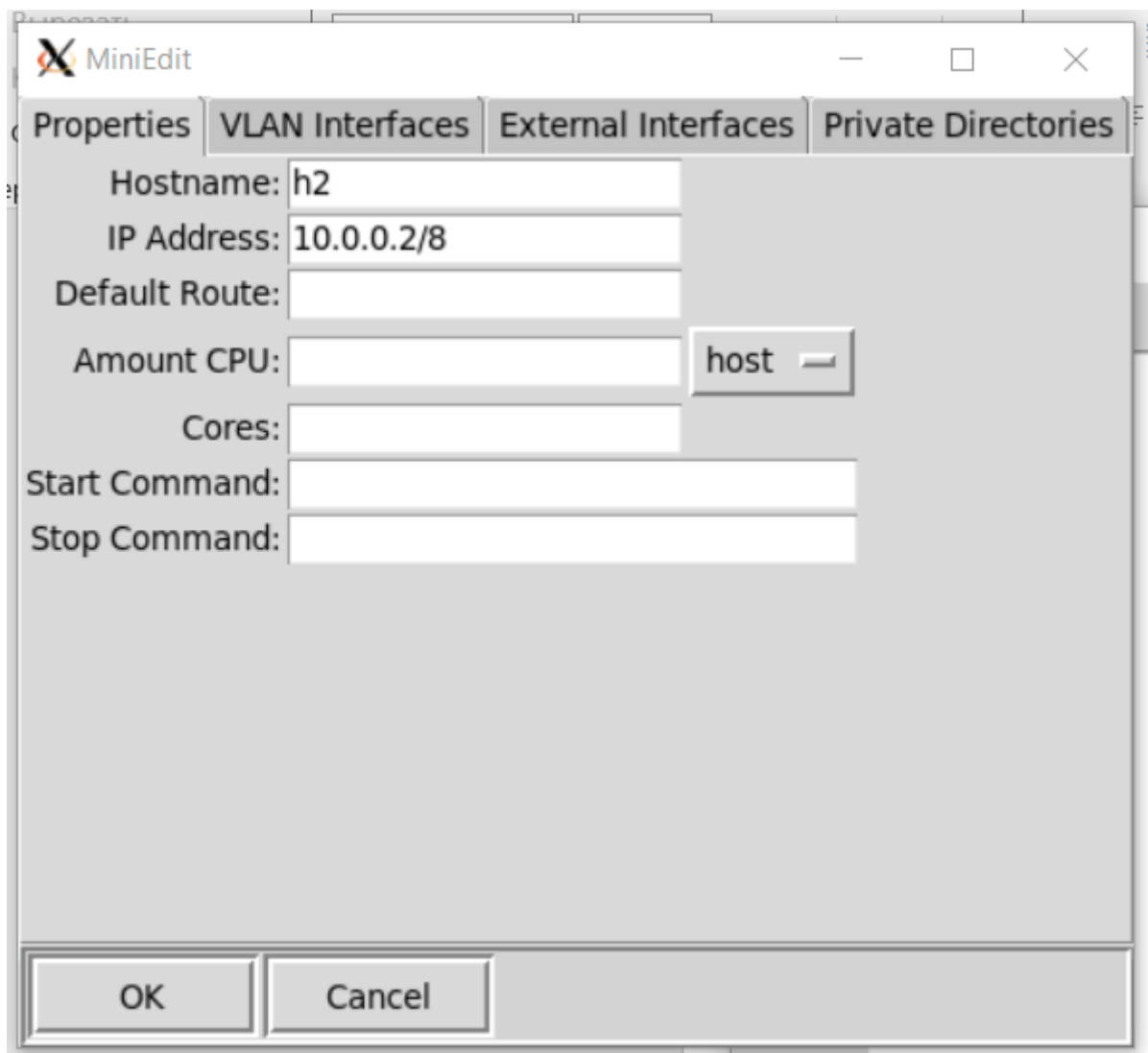
Amount CPU:

Cores:

Start Command:

Stop Command:

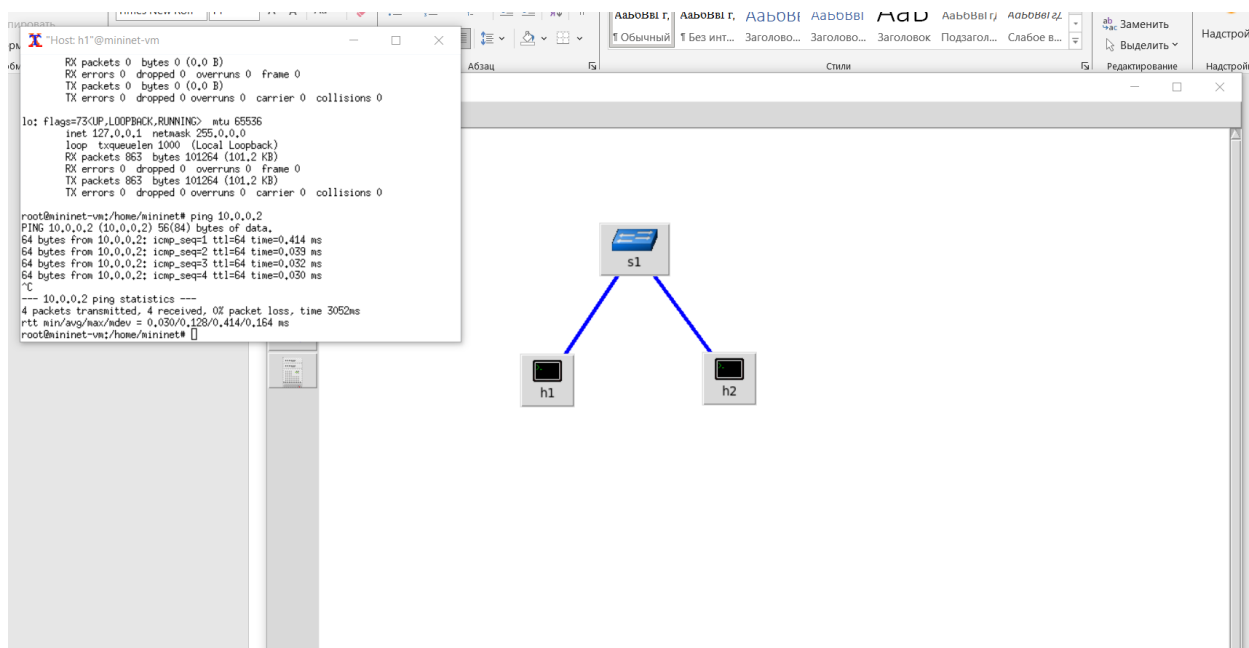
The form is a dialog box titled 'MiniEdit'. It has four tabs: 'Properties', 'VLAN Interfaces' (which is selected and highlighted in bold), 'External Interfaces', and 'Private Directories'. Below the tabs are several input fields and buttons. The 'Hostname' field contains 'h1'. The 'IP Address' field contains '10.0.0.1/8'. The 'Default Route' field is empty. The 'Amount CPU' field is empty, followed by a button labeled 'host'. The 'Cores' field is empty. There are two empty text input fields for 'Start Command' and 'Stop Command'. At the bottom are 'OK' and 'Cancel' buttons.



Проверка связности.

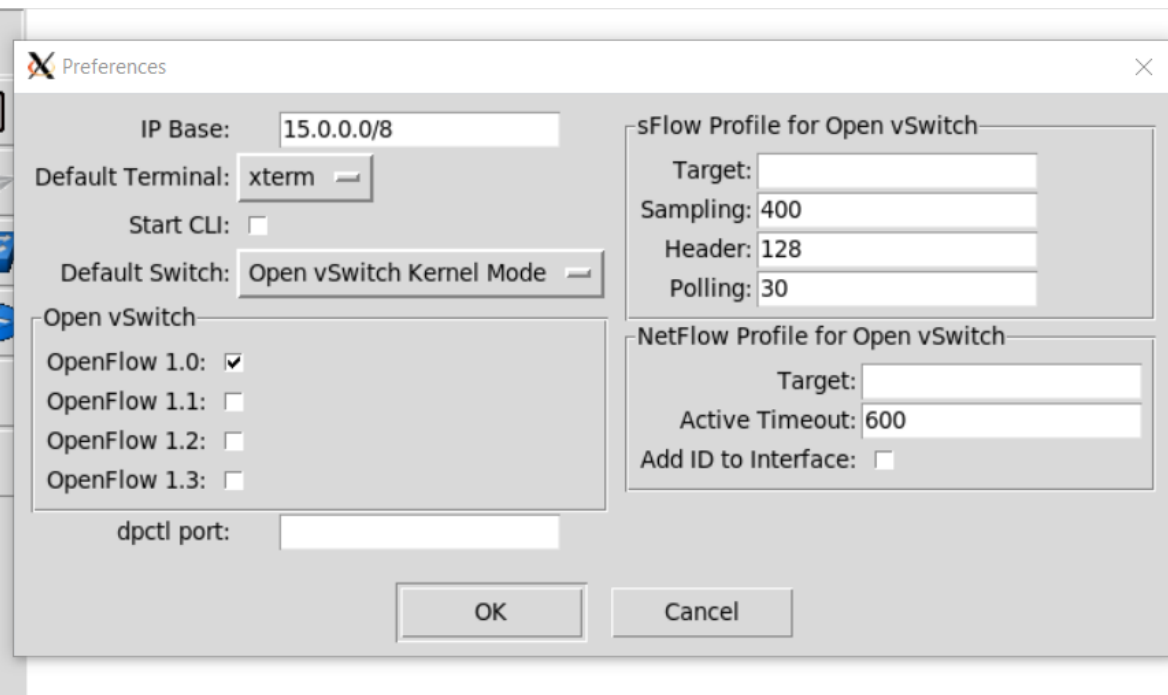
– Перед проверкой соединения между хостом h1 и хостом h2 необходимо запустить эмуляцию. Для запуска эмуляции нажмите кнопку Run. После начала эмуляции кнопки панели MiniEdit станут серыми, указывая на то, что в настоящее время они отключены. – Откройте терминал на хосте h1, удерживая правую кнопку мыши на хосте h1 и выбрав Terminal. Это действие позволит выполнять команды на хосте h1. – Откройте терминал на хосте h2. – На терминале хоста h1 введите 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. – Повторите 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. – Проверьте соединение между хостами, введя в терминале

хоста h1 команду `ping 10.0.0.2`. Для остановки теста нажмите `Ctrl + c`. – Остановите эмуляцию, нажав кнопку Stop.



Автоматическое назначение IP-адресов.

Ранее IP-адреса узлам h1 и h2 были назначены вручную. В качестве альтернативы можно полагаться на Mininet для автоматического назначения IP-адресов. – Удалите назначенный вручную IP-адрес с хостов h1 и h2. – В MiniEdit нажмите Edit Preferences . По умолчанию в поле базовые значения IP-адресов (IP Base) установлено 10.0.0.0/8. Измените это значение на 15.0.0.0/8. – Запустите эмуляцию, нажав кнопку Run. – Откройте терминал на хосте h1, удерживая правую кнопку мыши на хосте h1 и выбрав Terminal. – Чтобы отобразить IP-адреса, назначенные хосту h1, введите команду `ifconfig` Интерфейс h1-eth0 на узле h1 теперь имеет IP-адрес 15.0.0.1 и маску подсети 255.0.0.0.

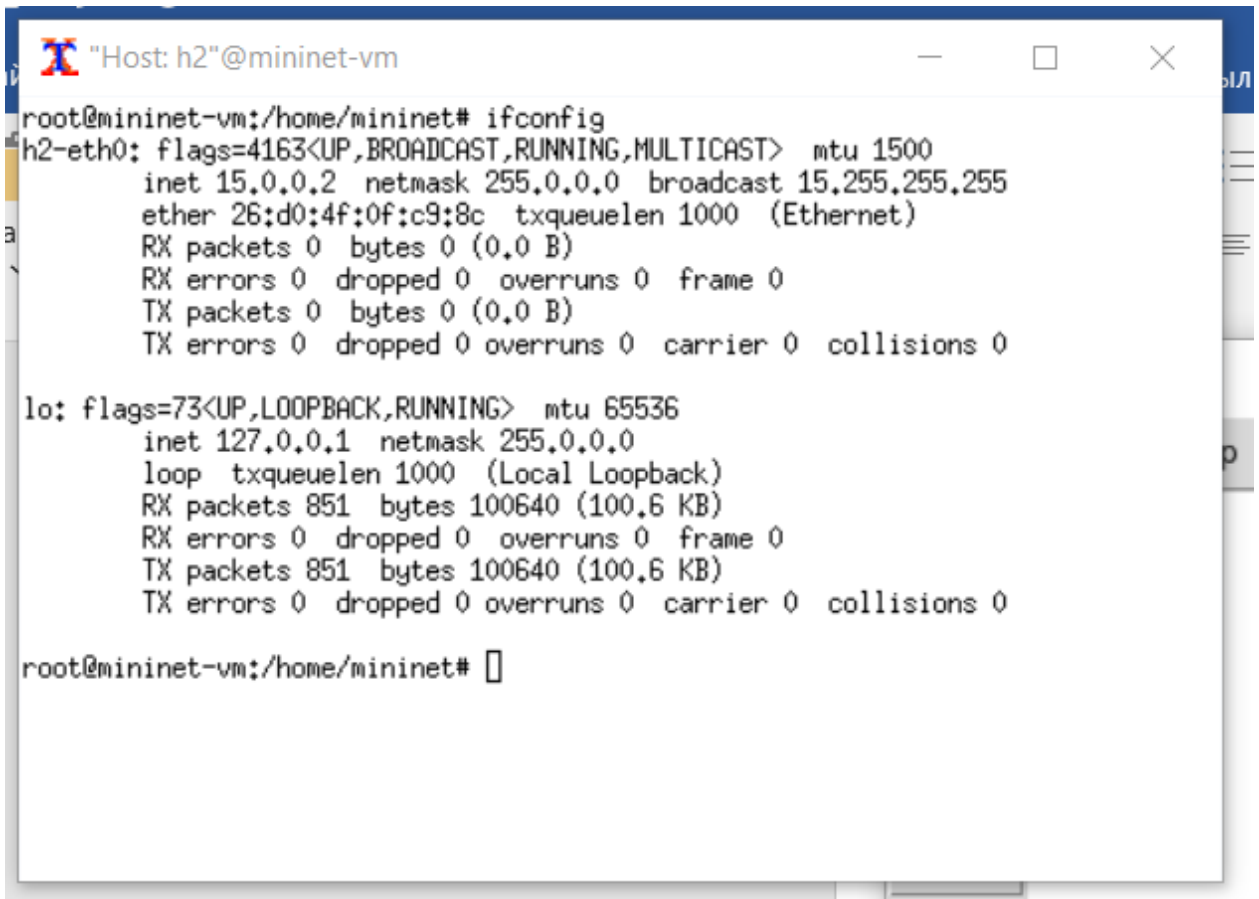


```
"Host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 15.0.0.1 netmask 255.0.0.0 broadcast 15.255.255.255
    ether 1a:de:da:19:a9:4c txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 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 650 bytes 90260 (90.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 650 bytes 90260 (90.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Вы также можете проверить IP-адрес, назначенный хосту h2. Соответствующий интерфейс h2-eth0 на хосте h2 должен иметь IP-адрес 15.0.0.2 и маску подсети 255.0.0.0. – Остановите эмуляцию, нажав кнопку Stop.



```
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 26:d0:4f:0f:c9:8c 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 851 bytes 100640 (100.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 851 bytes 100640 (100.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Сохранение и загрузка топологии Mininet.

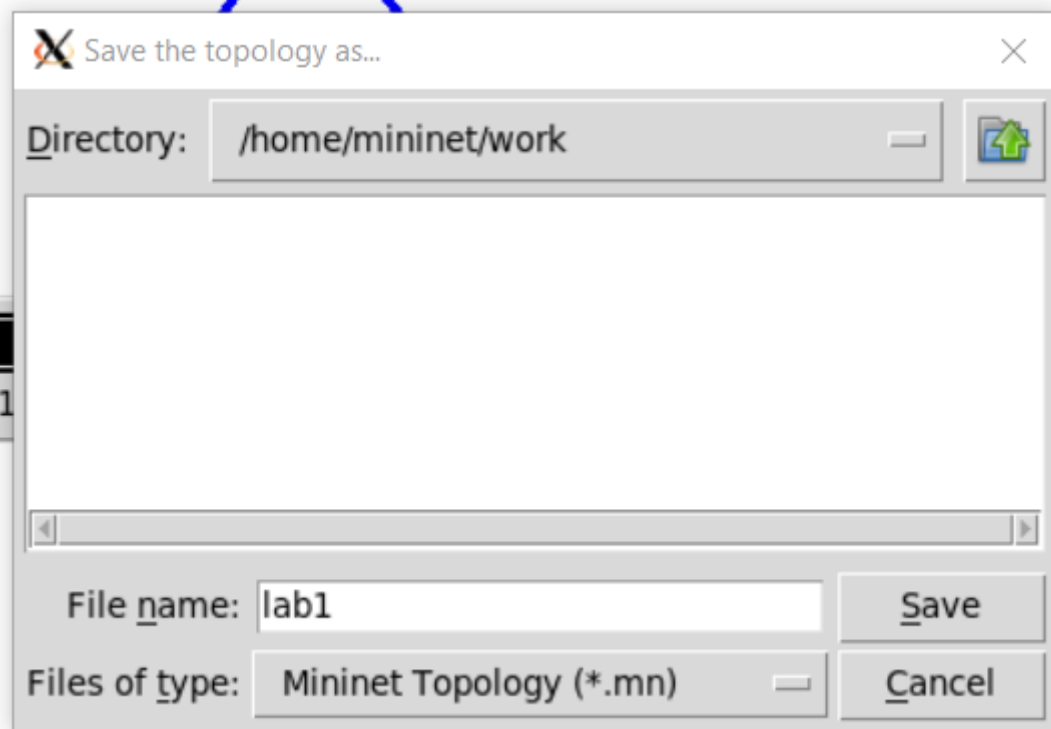
В домашнем каталоге виртуальной машины mininet создайте каталог для работы с проектами mininet:

```
mkdir ~/work
```



```
mininet@mininet-vm:~$ mkdir ~/work
```

Для сохранения топологии сети в файл нажмите в MiniEdit File Save . Укажите имя для топологии и сохраните на своём компьютере. – После сохранения проекта поменяйте права доступа к файлам в каталоге проекта: `sudo chown -R mininet:mininet ~/work`



После сохранения проекта поменяйте права доступа к файлам в каталоге проекта: `sudo chown -R mininet:mininet ~/work`

Для загрузки топологии в MiniEdit нажмите File Open .

```
a bytes-like object is required, not 'str'
mininet@mininet-vm:~$ sudo chown -R mininet:mininet ~/work
mininet@mininet-vm:~$
```

Вывод

Итогом лабораторной работы стало развёртывание в системе виртуализации (например, в VirtualBox) mininet, знакомство с основными командами для работы с Mininet через командную строку и через графический интерфейс.