

Министерство науки и высшего образования Российской Федерации
Федерального государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-
вычислительных систем (КИБЭВС)

ПРОГРАММНО-УПРАВЛЯЕМЫЕ СЕТИ. ЭМУЛЯТОР MININET

Отчет по лабораторной работе №3
по дисциплине «Сети и Системы Передачи Информации»

Выполнил

Студент гр. 728-2

_____ Геворгян Д.Р.

Принял

Доцент кафедры ТОР

_____ Агеев Е.Ю.

1 Введение

Целью лабораторной работы является анализ того, как происходит передача пакетов применительно к определённому коммутатору и узлам на примере эмулятора Mininet. Также дальнейшее исследование функционала программы Mininet и более углублённое изучение механизмов передачи и настройки соединения.

2 Ход работы

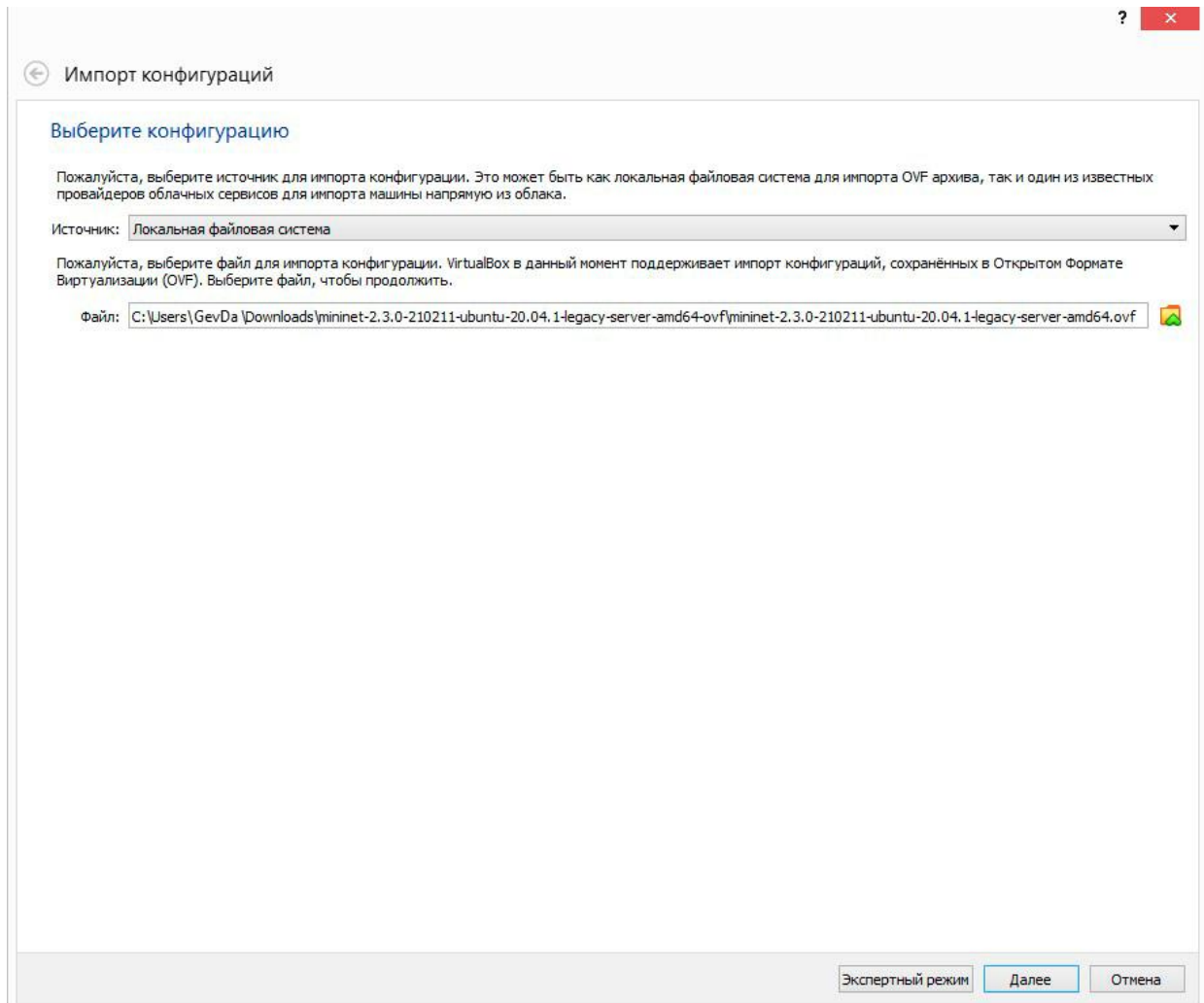


Рисунок 2.1 — Импорт машины

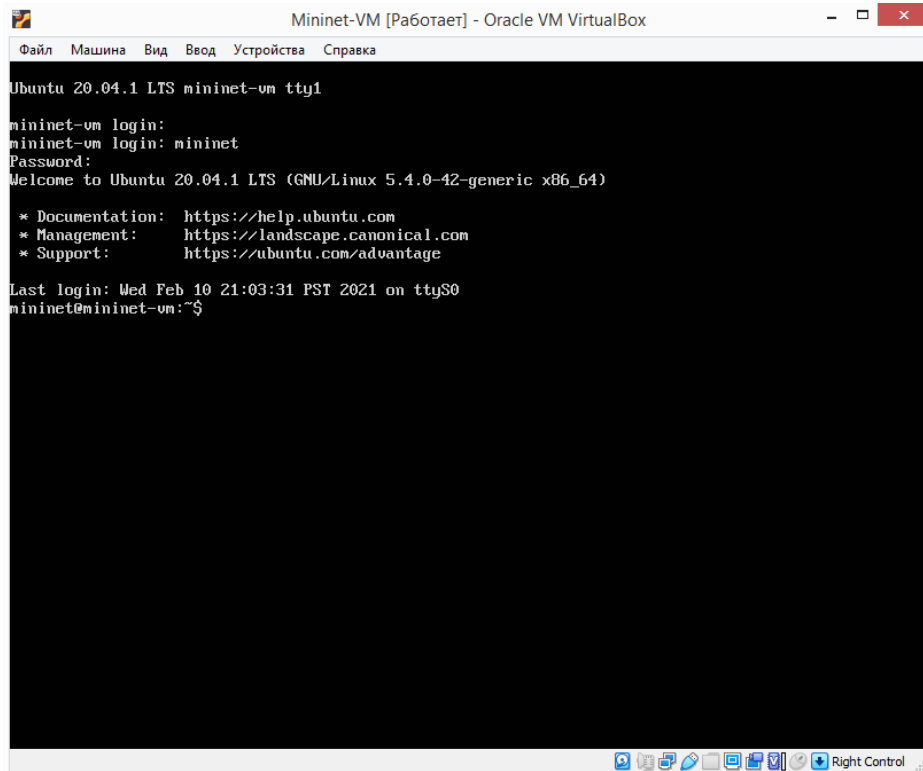


Рисунок 2.2 — Стартовое окно запуска

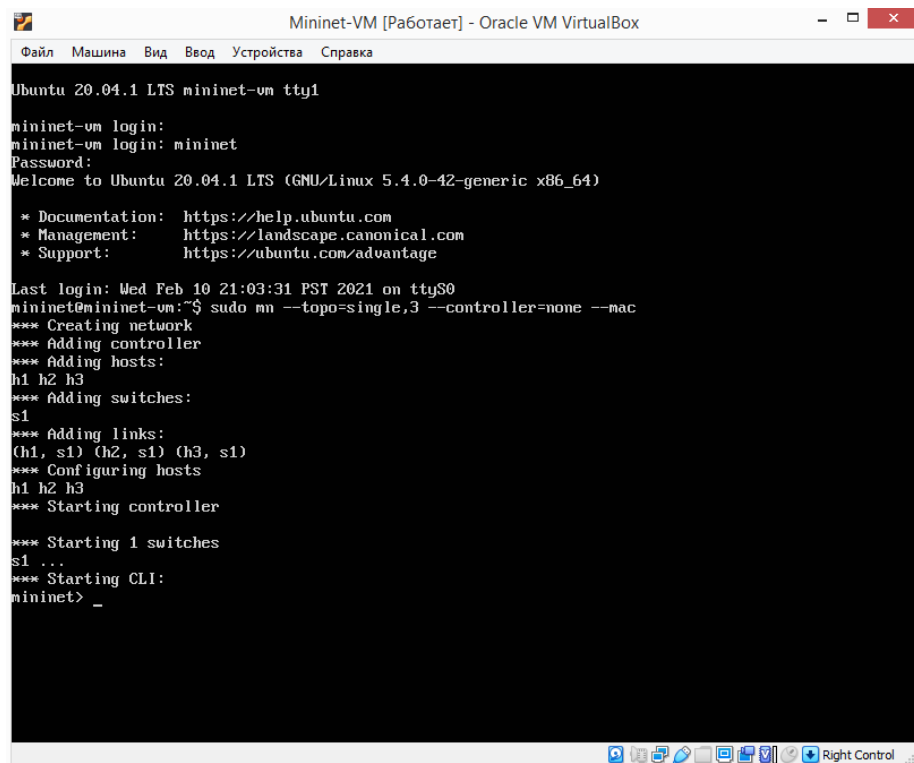


Рисунок 2.3 — Создание топологии

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1250>
<Host h2: h2-eth0:10.0.0.2 pid=1254>
<Host h3: h3-eth0:10.0.0.3 pid=1256>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=1261>
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0
mininet>
```

Рисунок 2.4 — Вывод информации о топологии

```
mininet> sh ovs-ofctl show s1
OFPPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000001
n_tables:254, n_buffers:0
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: output enqueue set_vlan_vid set_vlan_pcp strip_vlan mod_dl_src mod_dl_dst mod_nw_src mod_nw_dst mod_nw_tos mod_tp_src mod_tp_dst
1(s1-eth1): addr:0e:34:4e:5d:06:ba
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
2(s1-eth2): addr:76:38:5d:e7:3d:14
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
3(s1-eth3): addr:f2:58:01:b3:b3:9a
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(s1): addr:c6:ee:b1:41:0d:48
  config: PORT_DOWN
  state: LINK_DOWN
  speed: 0 Mbps now, 0 Mbps max
OFPPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
mininet>
```

Рисунок 2.5 — Вывод связей портов

```
OFPPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
mininet> sh ovs-ofctl add-flow s1 action=normal
mininet>
```

Рисунок 2.6 — Настройка обычного режима работы

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet>
```

Рисунок 2.7 — Проверка связи

```

Mininet-VM (После pingall) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
2021-05-13T22:47:17Z|100024|vconn|DBG|unix:/var/run/openvswitch/s1.mgmt: sent (Success): OFPT_HELLO (
xid=0x7):
  version bitmap: 0x01
2021-05-13T22:47:17Z|100025|vconn|DBG|unix:/var/run/openvswitch/s1.mgmt: received: OFPT_HELLO (OF1.5)
(xid=0xd):
  version bitmap: 0x01, 0x02, 0x03, 0x04, 0x05, 0x06
2021-05-13T22:47:17Z|100026|vconn|DBG|unix:/var/run/openvswitch/s1.mgmt: negotiated OpenFlow version
0x01 (we support version 0x01, peer supports version 0x06 and earlier)
2021-05-13T22:47:17Z|100027|vconn|DBG|unix:/var/run/openvswitch/s1.mgmt: sent (Success): OFPT_FEATURE
S_REQUEST (xid=0x8):
2021-05-13T22:47:17Z|100028|vconn|DBG|unix:/var/run/openvswitch/s1.mgmt: received: OFPT_FEATURES_REPL
Y (xid=0x8): dpid:0000000000000001
n_tables:254, n_buffers:0
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: output enqueue set_vlan_vid set_vlan_pcp strip_vlan mod_dl_src mod_dl_dst mod_nw_src mod_nw
dst mod_nw_tos mod_tp_src mod_tp_dst
1(s1-eth1): addr:0e:34:4e:5d:06:ba
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
2(s1-eth2): addr:76:38:5d:e7:3d:14
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
3(s1-eth3): addr:f2:58:01:b3:b3:9a
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(s1): addr:c6:ee:b1:41:0d:48
  config: PORT_DOWN
  state: LINK_DOWN
  speed: 0 Mbps now, 0 Mbps max
cookie=0x0, duration=146.329s, table=0, n_packets=24, n_bytes=1680, actions=NORMAL
mininet> _

```

Рисунок 2.8 — Вывод информации об обработке

```

mininet> sh ovs-ofctl del-flows s1
mininet> sh ovs-ofctl dump-flows s1
mininet> pingall
*** Ping: testing ping reachability
h1 -> X

```

Рисунок 2.9 — Удаление потоков

```

mininet> sh ovs-ofctl add-flow s1 priority=500,in_port=1,actions=output:2
mininet> sh ovs-ofctl add-flow s1 priority=500,in_port=2,actions=output:1
mininet> h1 ping -c2 h2
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=0.344 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.105 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1027ms
rtt min/avg/max/mdev = 0.105/0.224/0.344/0.119 ms
mininet> h3 ping -c2 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1022ms

mininet> _

```

Рисунок 2.10 — Проверка связи с первым и третьим узлами

```
mininet> sh ovs-ofctl dump-flows s1
cookie=0x0, duration=201.663s, table=0, n_packets=4, n_bytes=280, priority=500,in_port="s1-eth1" ac
tions=output:"s1-eth2"
cookie=0x0, duration=194.415s, table=0, n_packets=4, n_bytes=280, priority=500,in_port="s1-eth2" ac
tions=output:"s1-eth1"
mininet>
```

Рисунок 2.11 — Информация о потоках

```
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1024ms
pipe 2
mininet> h1 ping -c2 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1024ms

mininet> pingall
*** Ping: testing ping reachability
h1 -> X X
h2 -> X X
h3 -> X X
*** Results: 100% dropped (0/6 received)
mininet>
```

Рисунок 2.12 — «Дроп» пакетов

```
mininet> sh ovs-ofctl del-flows s1 --strict
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X
h2 -> h1 X
h3 -> X X
*** Results: 66% dropped (2/6 received)
mininet> h1 ping -c2 h2
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=0.243 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.107 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.107/0.175/0.243/0.068 ms
mininet> sh ovs-ofctl del-flows s1
mininet>
```

Рисунок 2.13 — Повторная проверка

```
mininet> sh ovs-ofctl del-flows s1
mininet> sh ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,actions=output:2
mininet> sh ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,actions=output:1
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X
h2 -> h1 X
h3 -> X X
*** Results: 66% dropped (2/6 received)
mininet> _
```

Рисунок 2.14 — Пересоздание потоков

```
mininet> sh ovs-ofctl add-flow s1 dl_type=0x806,nw_proto=1,action=flood
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X
h2 -> h1 X
h3 -> X X
*** Results: 66% dropped (2/6 received)
mininet>
```

Рисунок 2.15 — Добавление правила

```
mininet> sh ovs-ofctl del-flows s1
mininet> h3 python2 -m SimpleHTTPServer 80 &
mininet> sh ovs-ofctl add-flow s1 arp,actions=normal
mininet> sh ovs-ofctl add-flow s1 priority=500,dl_type=0x800,nw_proto=6,tp_dst=80,actions=output:3
mininet> sh ovs-ofctl add-flow s1
ovs-ofctl: 'add-flow' command requires at least 2 arguments
mininet> sh ovs-ofctl add-flow s1 priority=800,ip,nw_src=10.0.0.3,actions=normal
mininet> _
```

Рисунок 2.16 — Запуск сервера на хосте

```
mininet> h1 curl h3
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".gitconfig">.gitconfig</a>
<li><a href=".profile">.profile</a>
<li><a href=".sudo_as_admin_successful">.sudo_as_admin_successful</a>
<li><a href=".wget-hsts">.wget-hsts</a>
<li><a href=".wireshark/">.wireshark/</a>
<li><a href="mininet/">mininet</a>
<li><a href="oflops/">oflops</a>
<li><a href="oftest/">of test</a>
<li><a href="openflow/">openflow</a>
<li><a href="pox/">pox</a>
</ul>
<hr>
</body>
</html>
mininet> _
```

Рисунок 2.17 — Проверка обращения к серверу

3 Заключение

В результате выполнения лабораторной работы был выполнен анализ того, как происходит передача пакетов применительно к определённому коммутатору и узлам на примере эмулятора Mininet. Также произведено дальнейшее исследование функционала программы Mininet и более углублённое изучение механизмов передачи и настройки соединения.