

Отчет по лабораторной работе 5

Петрушов Дмитрий, 1032212287

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	15
	Список литературы	16

Список иллюстраций

2.1	Исправление прав запуска X-соединения в виртуальной машине mininet	6
2.2	Отображение информации их сетевых интерфейсов и IP-адресов	7
2.3	Проверка подключения между хостами h1 и h2	8
2.4	Добавление 10% потерь пакетов на хосте h1	8
2.5	Проверка	8
2.6	Добавление 10% потерь пакетов на хосте h2	9
2.7	Проверка	9
2.8	Восстановление конфигурации по умолчанию для хоста h1 и хоста h2	9
2.9	Добавление на узле h1 коэффициента потери пакетов 50%	10
2.10	Проверка	10
2.11	Восстановление конфигурации интерфейса по умолчанию	10
2.12	Добавление на узле h1 0.01% повреждения пакетов	11
2.13	Проверка конфигурации с помощью инструмента iPerf3 для проверки повторных передач	11
2.14	Добавление на узле h1 нового правила	11
2.15	Проверка	12
2.16	Добавление на узле h1 правила с дублированием 50% пакетов	12
2.17	Проверка	12
2.18	Создание каталога simple-drop и дальнейшее его открытие	13
2.19	Создание скрипта для эксперимента	13
2.20	Создание Makefile и помещение в него скрипта	14

Список таблиц

1 Цель работы

Основной целью работы является получение навыков проведения интерактивных экспериментов в среде Mininet по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных. Эти параметры влияют на производительность протоколов и сетей.

2 Выполнение лабораторной работы

В виртуальной машине mininet исправим права запуска X-соединения (рис. [2.1]):

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 ae1bcbbb82db3a3c878125643533139f
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 ae1bcbbb82db3a3c878125643533139f
root@mininet-vm:~# logout
mininet@mininet-vm:~$
```

Рис. 2.1: Исправление прав запуска X-соединения в виртуальной машине mininet

Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.

На хостах h1 и h2 введём команду `ifconfig`, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой `tc` будут использоваться интерфейсы h1-eth0 и h2-eth0 (рис. [2.2]):

```

root@mininet-vm:/home/mininet# 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 62:b8:48:67:cd:21 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 1328 bytes 579124 (579.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1328 bytes 579124 (579.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#

```

```

"host: h1" (на mininet-vm)
root@mininet-vm:/home/mininet# 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 2e:00:18:bc:98:98 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 1168 bytes 569212 (569.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1168 bytes 569212 (569.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 2.2: Отображение информации их сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -с 6 (рис. [2.3]):

```

root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=1.13 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.430 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.092 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5080ms
rtt min/avg/max/mdev = 0.049/0.311/1.133/0.389 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.3: Проверка подключения между хостами h1 и h2

На хосте h1 добавим 10% потерь пакетов к интерфейсу h1-eth0 (рис. [2.4]):

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 100
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

```

Рис. 2.4: Добавление 10% потерь пакетов на хосте h1

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 100 с хоста h1 (рис. [2.5]):

```

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 93 received, 7% packet loss, time 101340ms
rtt min/avg/max/mdev = 0.035/0.084/1.294/0.153 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.5: Проверка

Для эмуляции глобальной сети с потерей пакетов в обоих направлениях необходимо к соответствующему интерфейсу на хосте h2 также добавить 10% потерь пакетов (рис. [2.6]):


```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem loss 10 %
root@mininet-vm:/home/mininet#
```

Рис. 2.6: Добавление 10% потерь пакетов на хосте h2

Проверим, что соединение между хостом h1 и хостом h2 имеет больший процент потерянных данных (10% от хоста h1 к хосту h2 и 10% от хоста h2 к хосту h1), повторив команду ping с параметром -c 100 на терминале хоста h1 (рис. [2.7]):

```
--- 10.0.0.2 ping statistics ---
100 packets transmitted, 81 received, 19% packet loss, time 101330ms
rtt min/avg/max/mdev = 0.031/0.101/1.693/0.194 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.7: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. [2.8]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem loss 10 %
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet#

--- 10.0.0.2 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8197ms
rtt min/avg/max/mdev = 0.039/0.250/0.826/0.272 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.8: Восстановление конфигурации по умолчанию для хоста h1 и хоста h2

Убедимся, что соединение от хоста h1 к хосту h2 не имеет явной потери пакетов, запустив команду ping с терминала хоста h1 и затем нажав Ctrl + c , чтобы остановить тест.

Добавим на интерфейсе узла h1 коэффициент потери пакетов 50% (такой высокий уровень потери пакетов маловероятен), и каждая последующая вероятность зависит на 50% от последней (рис. [2.9]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 50% 50%
root@mininet-vm:/home/mininet#
```

Рис. 2.9: Добавление на узле h1 коэффициента потери пакетов 50%

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 50 с хоста h1 (рис. [2.10]):

```
50 packets transmitted, 35 received, 30% packet loss, time 50151ms
rtt min/avg/max/mdev = 0.038/0.153/1.321/0.262 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.10: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. [2.11]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 2.11: Восстановление конфигурации интерфейса по умолчанию

Добавим на интерфейсе узла h1 0,01% повреждения пакетов (рис. [2.12]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem corrupt
0.01%
root@mininet-vm:/home/mininet#
```

Рис. 2.12: Добавление на узле h1 0.01% повреждения пакетов

Проверим конфигурацию с помощью инструмента iPerf3 для проверки повторных передач. Для этого запустим iPerf3 в режиме сервера в терминале хоста h2, запустим iPerf3 в клиентском режиме в терминале хоста h1 (рис. [2.13]):

```
root@mininet-vm:/home/mininet# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 34270 connected to 10.0.0.2 port 5201
[ ID] Interval            Transfer          Bitrate          Retr  Cwnd
[ 7]  0.00-1.00    sec   5.07 GBytes    43.6 Gbits/sec    24   2.04 MBytes
[ 7]  1.00-2.00    sec   4.99 GBytes    42.8 Gbits/sec    26     704 KBytes
[ 7]  2.00-3.00    sec   4.97 GBytes    42.7 Gbits/sec    19     689 KBytes
[ 7]  3.00-4.00    sec   4.99 GBytes    42.9 Gbits/sec    22     700 KBytes
[ 7]  4.00-5.00    sec   4.90 GBytes    42.1 Gbits/sec    24     452 KBytes
[ 7]  5.00-6.00    sec   4.28 GBytes    36.8 Gbits/sec    22     675 KBytes
[ 7]  6.00-7.00    sec   4.66 GBytes    40.1 Gbits/sec    16     675 KBytes
[ 7]  7.00-8.00    sec   4.92 GBytes    42.3 Gbits/sec    30     561 KBytes
[ 7]  8.00-9.00    sec   4.93 GBytes    42.3 Gbits/sec    24     669 KBytes
[ 7]  9.00-10.00   sec   4.91 GBytes    42.2 Gbits/sec    27     669 KBytes
- - - - -
[ ID] Interval            Transfer          Bitrate          Retr
[ 7]  0.00-10.00   sec   48.6 GBytes    41.8 Gbits/sec    234
[ 7]  0.00-10.00   sec   48.6 GBytes    41.7 Gbits/sec
sender
receiver
```

Рис. 2.13: Проверка конфигурации с помощью инструмента iPerf3 для проверки повторных передач

Восстановим для узла h1 конфигурацию по умолчанию

Добавим на интерфейсе узла h1 нового правила (рис. [2.14]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
0ms reorder 25% 50%
```

Рис. 2.14: Добавление на узле h1 нового правила

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду `ping` с параметром `-c 20` с хоста h1. Убедимся, что часть пакетов не будут иметь задержки (один из четырех, или 25%), а последующие несколько пакетов будут иметь задержку около 10 миллисекунд (три из четырех, или 75%) (рис. [2.15]):

```
--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19028ms
rtt min/avg/max/mdev = 10.084/10.623/14.350/1.036 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.15: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1

Для интерфейса узла h1 зададим правило с дублированием 50% пакетов (т.е. 50% пакетов должны быть получены дважды) (рис. [2.16]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem duplicate 50%
root@mininet-vm:/home/mininet#
```

Рис. 2.16: Добавление на узле h1 правила с дублированием 50% пакетов

Проверим, что на соединении от хоста h1 к хосту h2 имеются дублированные пакеты, используя команду `ping` с параметром `-c 20` с хоста h1. Дубликаты пакетов помечаются как DUP!. Измеренная скорость дублирования пакетов будет приближаться к настроенной скорости по мере выполнения большего количества попыток (рис. [2.17]):

```
--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, +13 duplicates, 0% packet loss, time 19408ms
rtt min/avg/max/mdev = 0.039/0.163/1.491/0.293 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.17: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1.

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-drop и перейдём в него (рис. [2.18]):

```
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/simple-drop
mininet@mininet-vm:~$ cd ~/work/lab_netem_ii/simple-drop
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ touch lab_netem_ii.py
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$
```

Рис. 2.18: Создание каталога simple-drop и дальнейшее его открытие

Создадим скрипт для эксперимента lab_netem_ii.py (рис. [2.19]):

```
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info
import time

def emptyNet():
    "Create an empty network and add nodes to it."
    net = Mininet( controller=Controller, waitConnected=True )
    info( '*** Adding controller\n' )
    net.addController( 'c0' )

    info( '*** Adding hosts\n' )
```

Рис. 2.19: Создание скрипта для эксперимента

Создадим Makefile для управления процессом проведения эксперимента (рис. [2.20]):

```
all: ping.dat
ping.dat:
    sudo python lab_netem_i1.py
    sudo chown mininet:mininet ping.dat
clean:
    -rm -f *.dat
```

Рис. 2.20: Создание Makefile и помещение в него скрипта

Выполним эксперимент и далее очистим каталог от результатов проведения экспериментов.

Далее реализуем воспроизводимые эксперименты по исследованию параметров сети, связанных с потерей, изменением порядка и повреждением пакетов при передаче данных

3 Вывод

В ходе выполнения лабораторной работы получили навыки проведения интерактивных экспериментов в среде Mininet по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных.

Список литературы