

Отчет по лабораторной работе №5

Дисциплина: Моделирование сетей передачи данных

Лобанова Полина Иннокентьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	22
	Список литературы	23

Список иллюстраций

3.1	Изменение прав запуска X-соединения.	7
3.2	Создание топологии	8
3.3	Команда <code>ifconfig</code> на хосте <code>h1</code>	8
3.4	Команда <code>ifconfig</code> на хосте <code>h2</code>	8
3.5	Пингование	9
3.6	Добавление процента потерь	9
3.7	Пингование	10
3.8	Пингование	10
3.9	Добавление процента потерь	10
3.10	Пингование	11
3.11	Пингование	12
3.12	Восстановление конфигурации	12
3.13	Пингование	12
3.14	Добавление коэффициента потерь с корреляцией	13
3.15	Пингование	13
3.16	Добавление повреждения	13
3.17	Запуск сервера	14
3.18	Запуск клиента	14
3.19	Добавление переупорядочивания пакетов	14
3.20	Пингование	15
3.21	Добавление дублирования	15
3.22	Пингование	16
3.23	Создание каталога	16
3.24	Создание подкаталога	16
3.25	Скрипт <code>lab_netem_ii.py</code>	17
3.26	Изменение в скрипте <code>lab_netem_ii.py</code>	17
3.27	Создание <code>Makefile</code>	17
3.28	Выполнение эксперимента	18
3.29	Добавление коэффициента потерь с корреляцией	18
3.30	Выполнение эксперимента	19
3.31	Добавление повреждения	19
3.32	Выполнение эксперимента	19
3.33	Добавление переупорядочивания пакетов	20
3.34	Выполнение эксперимента	20
3.35	Добавление дублирования	20
3.36	Выполнение эксперимента	21

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 Задание

1. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных.
3. Реализуйте воспроизводимый эксперимент по добавлению правила отбрасывания пакетов в эмулируемой глобальной сети. На экран выведите сводную информацию о потерянных пакетах.
4. Самостоятельно реализуйте воспроизводимые эксперименты по исследованию параметров сети, связанных с потерей, изменением порядка и повреждением пакетов при передаче данных. На экран выведите сводную информацию о потерянных пакетах.

3 Выполнение лабораторной работы

1. Запустила виртуальную среду с mininet. Из основной ОС подключилась к виртуальной машине и исправила права запуска X-соединения. Скопировала значение куки своего пользователя mininet в файл для пользователя root.

```
[pilobanova@fedora ~]$ ssh -Y mininet@192.168.56.103
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

Last login: Sat Sep 13 06:29:15 2025 from 192.168.56.1
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  ecb597e678aeec4f6fed2fa4a884b3d2
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  ecb597e678a
eec4f6fed2fa4a884b3d2
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  ecb597e678a
eec4f6fed2fa4a884b3d2
root@mininet-vm:~# logout
mininet@mininet-vm:~$
```

Рис. 3.1: Изменение прав запуска X-соединения.

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

```

mininet@mininet-vm:~$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>

```

Рис. 3.2: Создание топологии

3. На хостах h1 и h2 ввела команду `ifconfig`, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам.

```

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 6a:3e:bf:e0:29:6e 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 1248 bytes 337844 (337.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1248 bytes 337844 (337.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 3.3: Команда `ifconfig` на хосте h1

```

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 46:11:1c:e7:71:5e 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 1176 bytes 331036 (331.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1176 bytes 331036 (331.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 3.4: Команда `ifconfig` на хосте h2

4. Проверила подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6. Минимальное значение RTT:0,031; среднее значение RTT: 0,3; максимальное значение RTT:1,37; стандартное отклонение: 0,48.

```
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=1.37 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.248 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.035 ms
```

Рис. 3.5: Пингование

5. На хосте h1 добавила 10% потерь пакетов к интерфейсу h1-eth0.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# █
```

Рис. 3.6: Добавление процента потерь

6. Проверила, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 100 с хоста h1.

```

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:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.723 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.265 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.322 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.130 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.026 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.043 ms

```

Рис. 3.7: Пингование

```

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 93 received, 7% packet loss, time 101385ms
rtt min/avg/max/mdev = 0.026/0.056/0.723/0.079 ms
root@mininet-vm:/home/mininet# █

```

Рис. 3.8: Пингование

7. Для эмуляции глобальной сети с потерей пакетов в обоих направлениях к соответствующему интерфейсу на хосте h2 также добавила 10% потерь пакетов.

```

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

```

Рис. 3.9: Добавление процента потерь

8. Проверила, 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. Отсутствующие из-за потери пакетов номера последовательности: 1, 3, 6, 16, 17, 24, 25, 29, 42, 44, 48, 51, 59, 61, 65, 67, 68, 72, 74, 75, 76, 77, 81, 86, 87, 99. Процент потерянных пакетов после завершения передачи: 26%.

```
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:
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.399 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.293 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.126 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.034 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=52 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=53 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=54 ttl=64 time=0.038 ms
```

Рис. 3.10: Пингование

```

64 bytes from 10.0.0.2: icmp_seq=55 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=56 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=57 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=58 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=60 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=62 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=63 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.042 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 74 received, 26% packet loss, time 1013
rtt min/avg/max/mdev = 0.033/0.050/0.399/0.051 ms

```

Рис. 3.11: Пингование

9. Восстановила конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1 и для получателя h2.

```

root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# 
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
root@mininet-vm:/home/mininet# 

```

Рис. 3.12: Восстановление конфигурации

10. Убедилась, что соединение от хоста h1 к хосту h2 не имеет явной потери пакетов, запустив команду ping с терминала хоста h1.

```

root@mininet-vm:/home/mininet# 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=0.639 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.309 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.137 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.038 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4077ms
rtt min/avg/max/mdev = 0.037/0.232/0.639/0.226 ms

```

Рис. 3.13: Пингование

11. Добавила на интерфейсе узла h1 коэффициент потери пакетов 50%, и каждая последующая вероятность зависит на 50% от последней.

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

Рис. 3.14: Добавление коэффициента потерь с корреляцией

12. Проверила, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 50 с хоста h1. Отсутствующие из-за потери пакетов номера последовательности: 2, 3, 4, 5, 7, 8, 9, 10, 15, 16, 19, 20, 21, 25, 26, 27, 28, 29, 30, 31, 32, 38, 39, 40, 41, 42, 43, 44, 50. Процент потерянных пакетов после завершения передачи: 58%. Восстановила для узла h1 конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса.

```
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 50
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=0.744 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.275 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.185 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.041 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 21 received, 58% packet loss, time 50183ms
rtt min/avg/max/mdev = 0.037/0.094/0.744/0.156 ms
root@mininet-vm:/home/mininet# █
```

Рис. 3.15: Пингование

13. Добавила на интерфейсе узла h1 0,01% повреждения пакетов.

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

Рис. 3.16: Добавление повреждения

14. Проверила конфигурацию с помощью инструмента iPerf3 для проверки повторных передач. Для этого запустила iPerf3 в режиме сервера в терминале хоста h2.

```
root@mininet-vm:/home/mininet# iperf3 -s
warning: this system does not seem to support IPv6 - trying IPv4
-----
Server listening on 5201
-----
```

Рис. 3.17: Запуск сервера

15. Запустите iPerf3 в клиентском режиме в терминале хоста h1.

```
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 51438 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec  3.23 GBytes  27.7 Gbits/sec   9   650 KBytes
[ 7]  1.00-2.00    sec  3.16 GBytes  27.1 Gbits/sec  12   665 KBytes
[ 7]  2.00-3.00    sec  3.17 GBytes  27.3 Gbits/sec   7   617 KBytes
[ 7]  3.00-4.00    sec  3.06 GBytes  26.3 Gbits/sec   6   611 KBytes
[ 7]  4.00-5.00    sec  3.08 GBytes  26.5 Gbits/sec   9   622 KBytes
[ 7]  5.00-6.00    sec  3.25 GBytes  27.9 Gbits/sec  11   584 KBytes
[ 7]  6.00-7.00    sec  3.25 GBytes  27.9 Gbits/sec   9   608 KBytes
[ 7]  7.00-8.00    sec  3.20 GBytes  27.5 Gbits/sec   6   631 KBytes
[ 7]  8.00-9.00    sec  3.28 GBytes  28.1 Gbits/sec   6   656 KBytes
[ 7]  9.00-10.00   sec  3.30 GBytes  28.4 Gbits/sec   3   653 KBytes
-----
[ ID] Interval      Transfer    Bitrate      Retr
[ 7]  0.00-10.00   sec  32.0 GBytes  27.5 Gbits/sec   78
[ 7]  0.00-10.00   sec  32.0 GBytes  27.4 Gbits/sec
iperf Done.
sender
receiver
```

Рис. 3.18: Запуск клиента

16. Восстановила для узла h1 конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса.
17. Добавила на интерфейсе узла h1 следующее правило. Здесь 25% пакетов (со значением корреляции 50%) будут отправлены немедленно, а остальные 75% будут задержаны на 10 мс.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 10ms reorder 25% 50%
root@mininet-vm:/home/mininet# █
```

Рис. 3.19: Добавление переупорядочивания пакетов

18. Проверила, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду `ping` с параметром `-c 20` с хоста h1. Отсутствующих из-за потери пакетов номеров последовательности нет. Процент потерянных пакетов после завершения передачи: 0%. Восстановила конфигурацию интерфейса по умолчанию на узле h1.

```
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 20
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=0.780 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.230 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.120 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=10.9 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=10.7 ms
--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19094ms
rtt min/avg/max/mdev = 0.120/9.188/10.856/3.703 ms
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 20
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
```

Рис. 3.20: Пингование

19. Для интерфейса узла h1 задала правило с дублированием 50% пакетов (т.е. 50% пакетов должны быть получены дважды).

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

Рис. 3.21: Добавление дублирования

20. Проверила, что на соединении от хоста h1 к хосту h2 имеются дублированные пакеты, используя команду `ping` с параметром `-c 20` с хоста h1. Восстановила конфигурацию интерфейса по умолчанию на узле h1.

```

root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 20
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=0.880 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.985 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.212 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.138 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.058 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.041 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.093 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.068 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.039 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.043 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.040 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.039 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.037 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, +9 duplicates, 0% packet loss, time 19435ms
rtt min/avg/max/mdev = 0.037/0.118/0.985/0.224 ms

```

Рис. 3.22: Пингование

21. Для каждого воспроизводимого эксперимента ехрname создала свой каталог, в котором будут размещаться файлы эксперимента.

```

mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/exprname
mininet@mininet-vm:~$

```

Рис. 3.23: Создание каталога

22. В виртуальной среде mininet в своём рабочем каталоге с проектами создала каталог simple-drop и перешла в него.

```

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$ ls
lab_netem_ii.py

```

Рис. 3.24: Создание подкаталога

23. Создала скрипт для эксперимента lab_netem_ii.py.


```

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' )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( '*** Adding switch\n' )
    s1 = net.addSwitch( 's1' )

    info( '*** Creating links\n' )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )

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

    info( '*** Set delay\n' )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem loss 10%' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem loss 10%' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'/s/time=//g\' -e \'/icmp_seq=//g\' > ping.dat' )

    info( '*** Stopping network' )
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()

```

Рис. 3.25: Скрипт *lab_netem_ii.py*

24. Скорректировала скрипт так, чтобы в отдельный файл выводилась информация о потерях пакетов.

```

info( '*** Ping\n' )
h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "packet loss" | awk \'{print $6, $7, $8}\'} > ping.dat' )

```

Рис. 3.26: Изменение в скрипте *lab_netem_ii.py*

25. Создала Makefile для управления процессом проведения эксперимента.

```

all: ping.dat

ping.dat:
    sudo python lab_netem_ii.py
    sudo chown mininet:mininet ping.dat

clean:
    -rm -f *.dat

```

Рис. 3.27: Создание *Makefile*

26. Выполнила эксперимент.

```
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem loss 10%',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem loss 10%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "packet loss" | awk \'{print $6, $7, $8}\'} > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ cat ping.dat
15% packet loss,
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$
```

Рис. 3.28: Выполнение эксперимента

27. Самостоятельно реализовала воспроизводимые эксперименты по исследованию параметров сети, связанных с потерей пакетов и добавление значения корреляции для потери пакетов.

```
info( '*** Set delay\n')
h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem loss 50% 50%  
#h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem loss 10%' )
```

Рис. 3.29: Добавление коэффициента потерь с корреляцией

```

mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem loss 50% 50%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "packet loss" | awk \'{print $6, $7, $8}\'} > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ cat ping.dat
49% packet loss,

```

Рис. 3.30: Выполнение эксперимента

28. Самостоятельно реализовала воспроизводимые эксперименты по исследованию параметров сети, связанных добавлением повреждения пакетов.

```

mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem corrupt 0.01%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "packet loss" | awk \'{print $6, $7, $8}\'} > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ cat ping.dat
0% packet loss,

```

Рис. 3.31: Добавление повреждения

```

mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem corrupt 0.01%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "packet loss" | awk \'{print $6, $7, $8}\'} > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ cat ping.dat
0% packet loss,

```

Рис. 3.32: Выполнение эксперимента

29. Самостоятельно реализовала воспроизводимые эксперименты по исследованию параметров сети, связанных добавлением переупорядочивания пакетов.

```
info( '*** Set delay\n')
h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 10ms reorder 25% 50%' )
#h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem loss 10%' )
```

Рис. 3.33: Добавление переупорядочивания пакетов

```
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem delay 10ms reorder 25% 50%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "packet loss" | awk \'{print $6, $7, $8}\\' > ping.dat}')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ cat ping.dat
0% packet loss,
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$
```

Рис. 3.34: Выполнение эксперимента

30. Самостоятельно реализовала воспроизводимые эксперименты по исследованию параметров сети, связанных добавлением дублирования пакетов.

```
info( '*** Set delay\n')
h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem duplicate 50%' )
#h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem loss 10%' )
```

Рис. 3.35: Добавление дублирования

```

mininet@mininet-vm: /work/lab_netem_ii/simple_drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem duplicate 50%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "packet loss" | awk \'{print $6, $7, $8}\'} > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm: /work/lab_netem_ii/simple_drop$ cat ping.dat
+58 duplicates, 0%

```

Рис. 3.36: *Выполнение эксперимента*

4 Выводы

Я получила 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