

Лабораторная работа №5

Эмуляция и измерение потерь пакетов в глобальных сетях

Барабанова Кристина

Российский университет дружбы народов, Москва, Россия

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.

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.

Выполнение лабораторной работы

1. Запуск лабораторной топологии

1. Запустите виртуальную среду с mininet.
2. Из основной ОС подключитесь к виртуальной машине:

```
mininet@mininet-vm:~$ ssh -Y mininet@192.168.56.104
mininet@192.168.56.104'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: Fri Jan 30 10:45:15 2026 from 192.168.56.1
```

Рис. 1: рис. 1.1

3. В виртуальной машине mininet при необходимости исправьте права запуска X-соединения. Скопируйте значение куки (MIT magic cookie)1 своего пользователя mininet в файл для пользователя root:

```
mininet@mininet-vm:~$ mininet@mininet-vm:~$ xauth list $DISPLAY
mininet@mininet-vm:~$: command not found
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:11  MIT-MAGIC-COOKIE-1  08e502f854e407c16d854e2ae87d7b37
mininet@mininet-vm:~$ ^C
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/ unix:11  MIT-MAGIC-COOKIE-1  08e502f854e
407c16d854e2ae87d7b37
root@mininet-vm:~# ldstout
ldstout: command not found
root@mininet-vm:~# logout
```

Рис. 2: рис. 1.2

4. Задайте 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:11.0
*** Starting controller
c0
*** Starting 1 s
s1 ...
*** Starting CLI
mininet> 
```

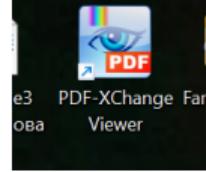
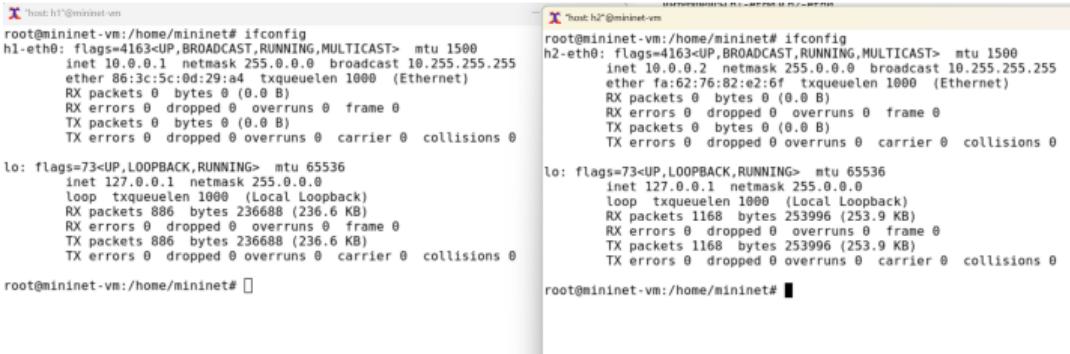


Рис. 3: рис. 1.3

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



The image shows two terminal windows side-by-side. The left window is titled "host: h1" and the right window is titled "host: h2". Both windows are running the command "ifconfig".

Host h1:

```
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 86:3c:5c:0d:29:a4 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 886 bytes 236688 (236.6 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 886 bytes 236688 (236.6 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Host h2:

```
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 fa:62:76:82:e2:6f 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 253996 (253.9 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 1168 bytes 253996 (253.9 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 4: рис. 1.4

6. Проверьте подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6.

```
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=2.10 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.250 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.066 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5092ms
rtt min/avg/max/mdev = 0.054/0.439/2.100/0.745 ms
root@mininet-vm:/home/mininet# █
```

Рис. 5: рис. 1.5

2. Добавление потери пакетов на интерфейс, подключённый к эмулируемой глобальной сети

1. На хосте h1 добавьте 10% потерь пакетов к интерфейсу h1-eth0:

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

Рис. 6: рис. 2.1

2. Проверьте, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -с 100 с хоста h1. Параметр -с указывает общее количество пакетов для отправки.

```
X "host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.110 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.061 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.071 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 89 received, 11% packet loss, time 101335ms
rtt min/avg/max/mdev = 0.047/0.100/1.500/0.159 ms
```

Рис. 7: рис. 2.2

3. Для эмуляции глобальной сети с потерей пакетов в обоих направлениях необходимо к соответствующему интерфейсу на хосте h2 также добавить 10% потерь пакетов:

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

Рис. 8: рис. 2.3

4. Проверьте, 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.

```
root@host-h1:~# ping -c 100 10.0.0.2
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=0.163 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.152 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.121 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.066 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 79 received, 21% packet loss, time 101334ms
rtt min/avg/max/mdev = 0.050/0.097/1.107/0.124 ms
root@mininet-vm:/home/mininet#
```

5. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1:

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

Рис. 10: рис. 2.5

Для получателя h2:

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

Рис. 11: рис. 2.6

6. Убедитесь, 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 -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=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.392 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.293 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.053 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=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.066 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=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.057 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19424ms
rtt min/avg/max/mdev = 0.044/0.142/1.111/0.238 ms
```

Рис. 12: рис. 2.7

3. Добавление значения корреляции для потери пакетов в эмулируемой глобальной сети

- Добавьте на интерфейсе узла h1 коэффициент потери пакетов 50% (такой высокий уровень потери пакетов маловероятен), 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# ping 10.0.0.2 -c 50
```

Рис. 13: рис. 3.1

2. Проверьте, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -с 50 с хоста h1.

```
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=1.40 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.347 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.149 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.257 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.142 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 24 received, 52% packet loss, time 50137ms
rtt min/avg/max/mdev = 0.047/0.143/1.402/0.271 ms
```

4. Добавление повреждения пакетов в эмулируемой глобальной сети

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Добавьте 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%
```

Рис. 15: рис. 4.1

3. Проверьте конфигурацию с помощью инструмента iPerf3 для проверки повторных передач. Для этого: – запустите iPerf3 в режиме сервера в терминале хоста h2:

```
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 42340 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00   sec   2.79 GBytes   24.0 Gbits/sec   18  782 KBytes
[ 7]  1.00-2.00   sec   2.69 GBytes   23.1 Gbits/sec    6  943 KBytes
[ 7]  2.00-3.00   sec   2.55 GBytes   21.9 Gbits/sec    5  157 KBytes
[ 7]  3.00-4.00   sec   2.66 GBytes   22.8 Gbits/sec    7  525 KBytes
[ 7]  4.00-5.00   sec   2.74 GBytes   23.5 Gbits/sec    8  782 KBytes
[ 7]  5.00-6.00   sec   2.66 GBytes   22.8 Gbits/sec    5  130 KBytes
[ 7]  6.00-7.00   sec   2.82 GBytes   24.3 Gbits/sec    8  250 KBytes
[ 7]  7.00-8.00   sec   2.56 GBytes   22.0 Gbits/sec    8  814 KBytes
[ 7]  8.00-9.00   sec   2.64 GBytes   22.7 Gbits/sec    9  1.53 MBytes
[ 7]  9.00-10.00  sec   2.68 GBytes   23.0 Gbits/sec   3  602 KBytes
[ ID] Interval          Transfer     Bitrate      Retr
[ 7]  0.00-10.00  sec   26.8 GBytes   23.0 Gbits/sec   77
[ 7]  0.00-10.00  sec   26.8 GBytes   23.0 Gbits/sec
iperf Done.
```

Рис. 16: рис. 4.2

- 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
```

5. Добавление переупорядочивания пакетов в интерфейс подключения к эмулируемой глобальной сети

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Добавьте на интерфейсе узла h1 следующее правило.
3. Проверьте, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 20 с хоста h1. Убедитесь, 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%).

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
0ms reorder 25% 50%
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=10.6 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.152 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=10.2 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=10.5 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.2 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=10.1 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=10.1 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=10.5 ms
```

6. Добавление дублирования пакетов в интерфейс подключения к эмулируемой глобальной сети

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Для интерфейса узла h1 задайте правило с дублированием 50% пакетов (т.е. 50% пакетов должны быть получены дважды):
3. Проверьте, 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.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem duplicate 50%
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=1.22 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.25 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.374 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.794 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.153 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.054 ms
```

7. Воспроизведение экспериментов

1. Для каждого воспроизводимого эксперимента `expname` создайте свой каталог, в котором будут размещаться файлы эксперимента:

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

Рис. 20: рис. 7.1

1. В виртуальной среде mininet в своём рабочем каталоге с проектами создайте каталог simple-drop и перейдите 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
```

Рис. 21: рис. 7.2

2. Создаёте скрипт для эксперимента lab_netem_ii.py:

```
GNU nano 4.8                                lab_netem_ii.py
#!/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" )
    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,>'

    info( "*** Stopping network" )
    net.stop()
```

4. Скорректируйте скрипт так, чтобы на экран или в отдельный файл выводилась информация о потерях пакетов.

```
GNU nano 4.8          lab netem ii.py
#!/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' )
    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( 'fping -c 100 {h2.IP()} | tee ping_full.txt' |
        '| grep "time=" ')
```

5. Создайте Makefile для управления процессом проведения эксперимента:

```
GNU nano 4.8                                Makefile
all: ping.dat

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

clean:
    -rm -f *.dat
```

Рис. 24: рис. 7.5

6. Выполните 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 | tee ping_full.txt | grep "time=" | awk \'{priorit $5, $7}\' | sed -e \'s/time//g\' -e \'s/icmp_seq//g\' > 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
```

Рис. 25: рис. 7.6

7. Очистите каталог от результатов проведения экспериментов:

```
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ make clean  
rm -f *.dat
```

Рис. 26: рис. 7.7

Вывод

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.