Лабораторная работа №5

Эмуляция и измерение потерь пакетов в глобальных сетях

Хрусталев Влад Николаевич

Содержание

Список иллюстраций

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 Теоретическое введение

Mininet[1] – это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины(ВМ). На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(ifconfig, ping) и даже получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик.

3 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 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.

4 Выполнение лабораторной работы

В виртуальной машине mininet исправим права запуска X-соединения (рис. 4.1).

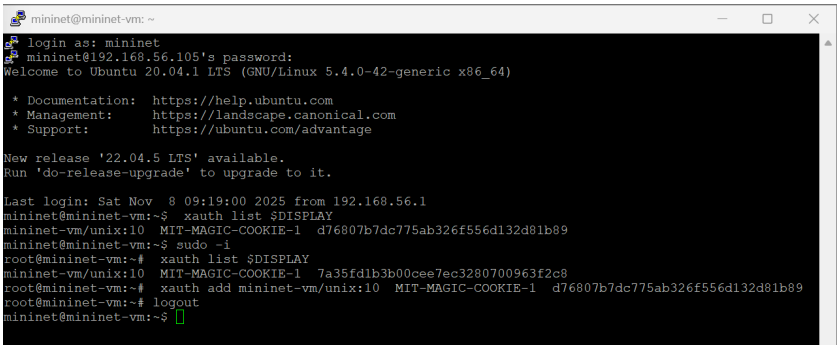

Рис. 4.1: Исправление прав запуска X-соединения в виртуальной машине mininet

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 введем 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. 4.2).

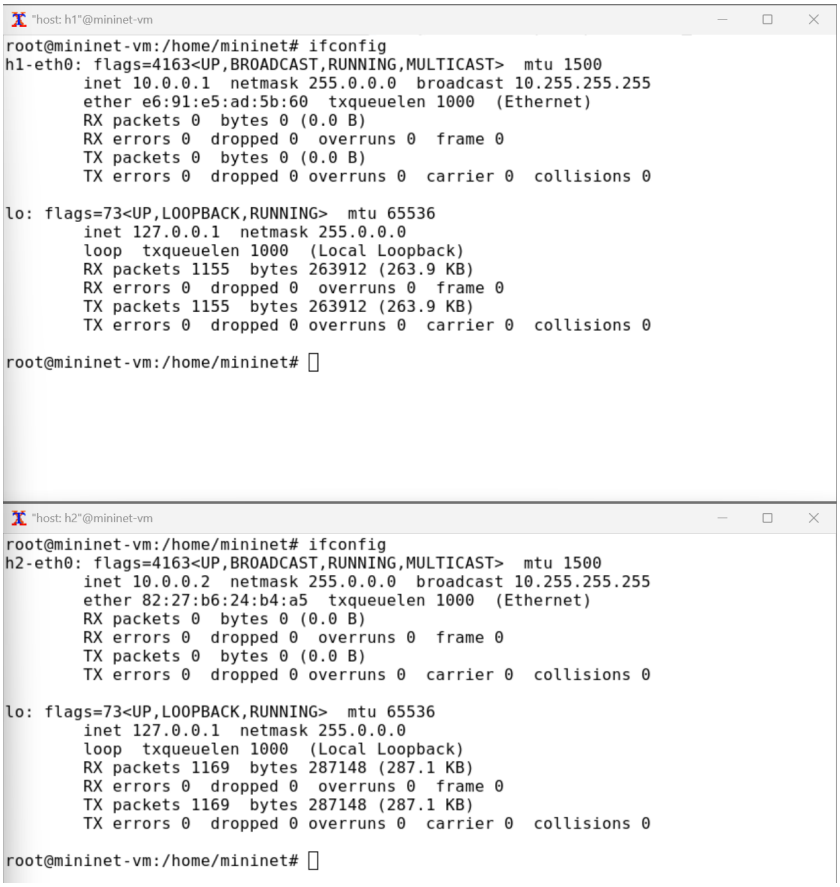

Рис. 4.2: Информацию о сетевых интерфейсах и IP-адресах хостов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. 4.3).

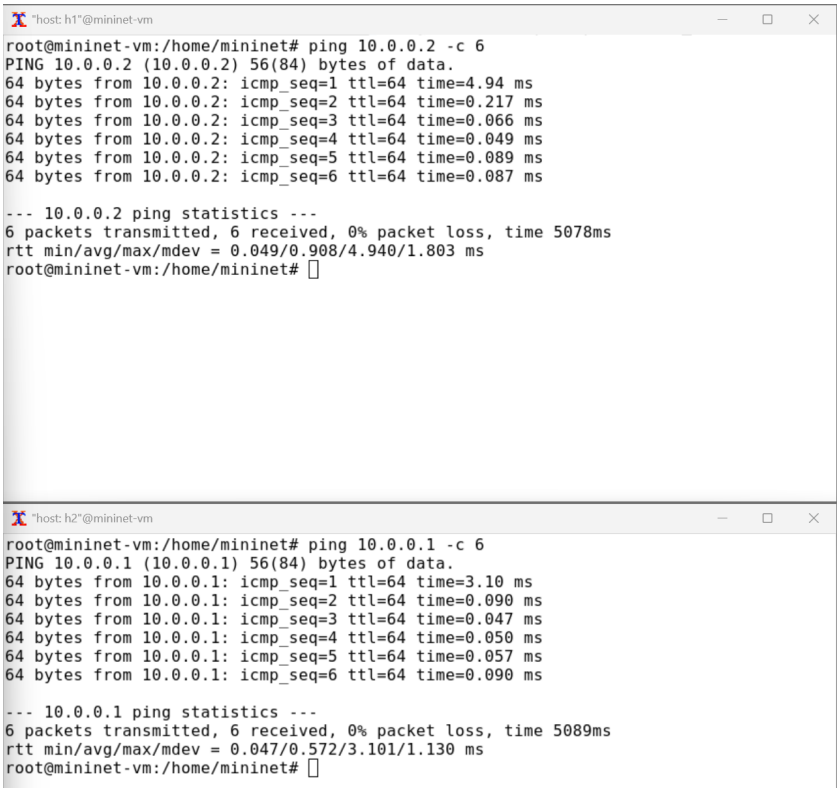

Рис. 4.3: Проверка подключения между хостами h1 и h2

Пакеты могут быть потеряны в процессе передачи из-за таких факторов, как битовые ошибки и перегрузка сети. Скорость потери данных часто измеряется как процентная доля потерянных пакетов по отношению к количеству отправленных пакетов. На хосте h1 добавим 10% потерь пакетов к интерфейсу h1-eth0 (рис. 4.4):

```
sudo tc qdisc add dev h1-eth0 root netem loss 10%
```

Здесь:

- sudo: выполнить команду с более высокими привилегиями;
- tc: вызвать управление трафиком Linux;
- qdisc: изменить дисциплину очередей сетевого планировщика;
- add: создать новое правило;
- dev h1-eth0: указать интерфейс, на котором будет применяться правило;
- netem: использовать эмулятор сети;

- loss 10%: 10% потерь пакетов.

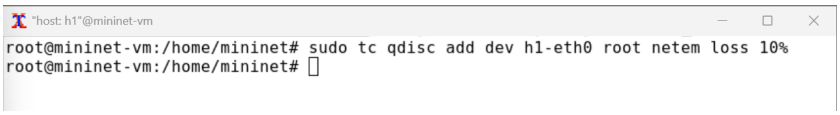

Рис. 4.4: Добавление 10% потерь пакетов на хосте h1

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 100 с хоста h1.(рис. 4.5).

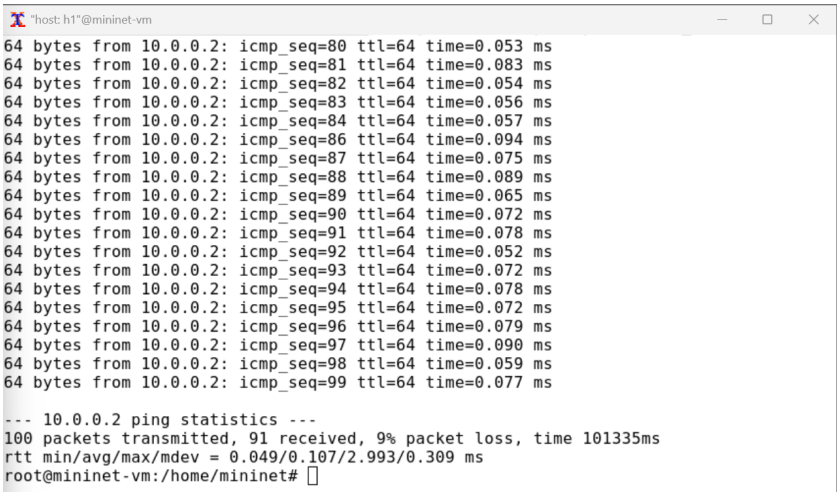

Рис. 4.5: Проверка потерь пакетов 1

Для эмуляции глобальной сети с потерей пакетов в обоих направлениях необходимо к соответствующему интерфейсу на хосте h2 также добавить 10% потерь пакетов. 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. 4.6).

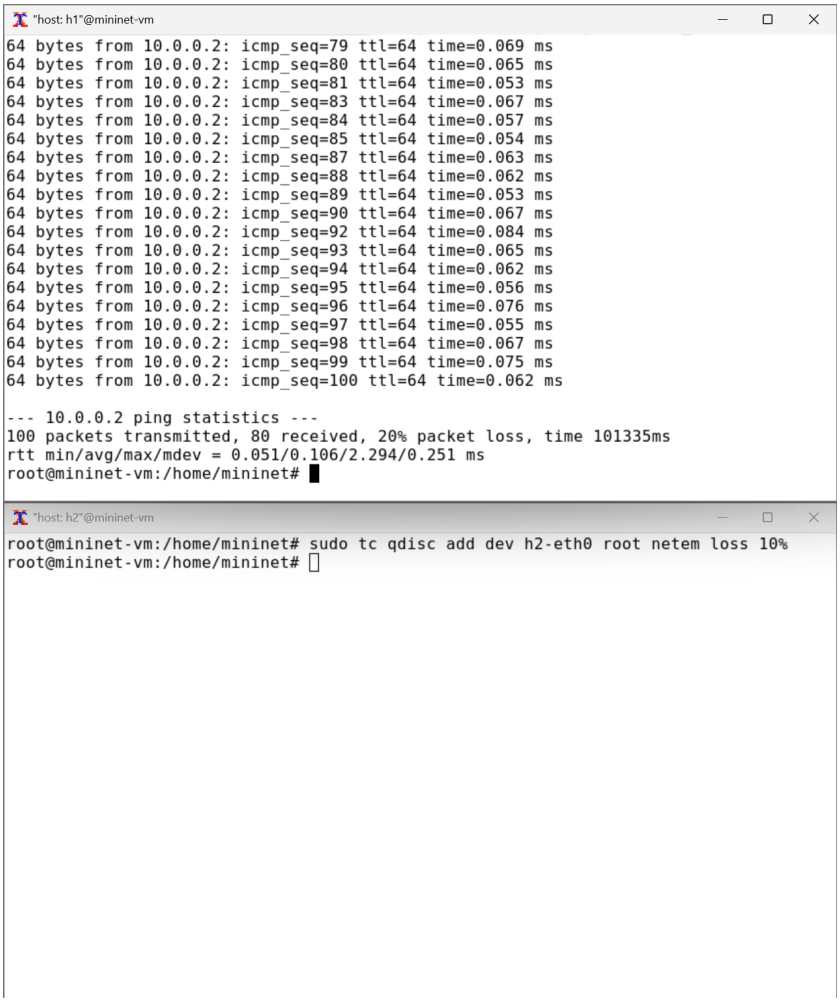

Рис. 4.6: Проверка потерь пакетов 2

Пропущенны номера icmp_seq (возможны значения от 1 до 100): [2, 9, 14, 19, 22, 25, 31, 33, 42, 44, 49, 55, 59, 67, 69, 73, 78, 82, 85, 91].

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. 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(рис. 4.7).

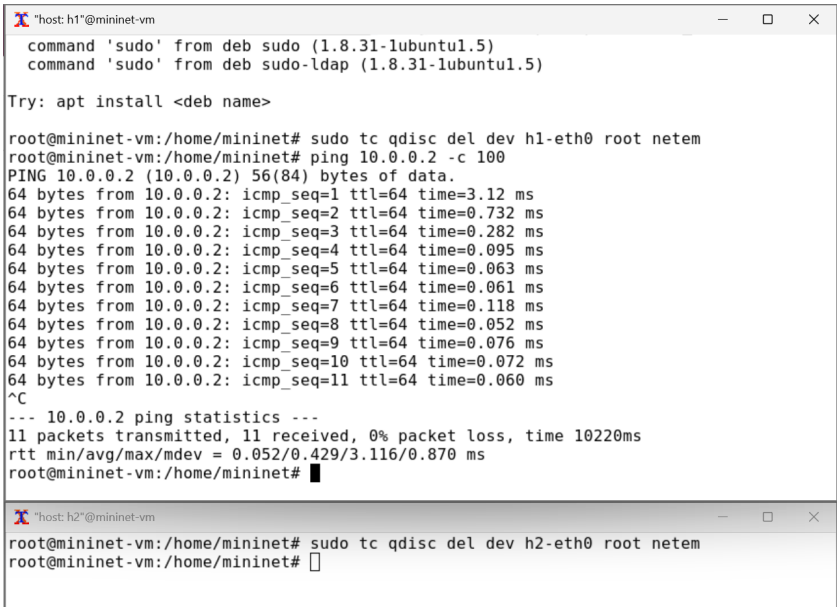

Рис. 4.7: Восстановление конфигурации по умолчанию для хоста h1 и хоста h2 и проверка

Добавим на интерфейсе узла h1 коэффициент потери пакетов 50% (такой высокий уровень потери пакетов маловероятен), и каждая последующая вероятность зависит на 50% от последней: Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 50 с хоста h1(рис. 4.8).

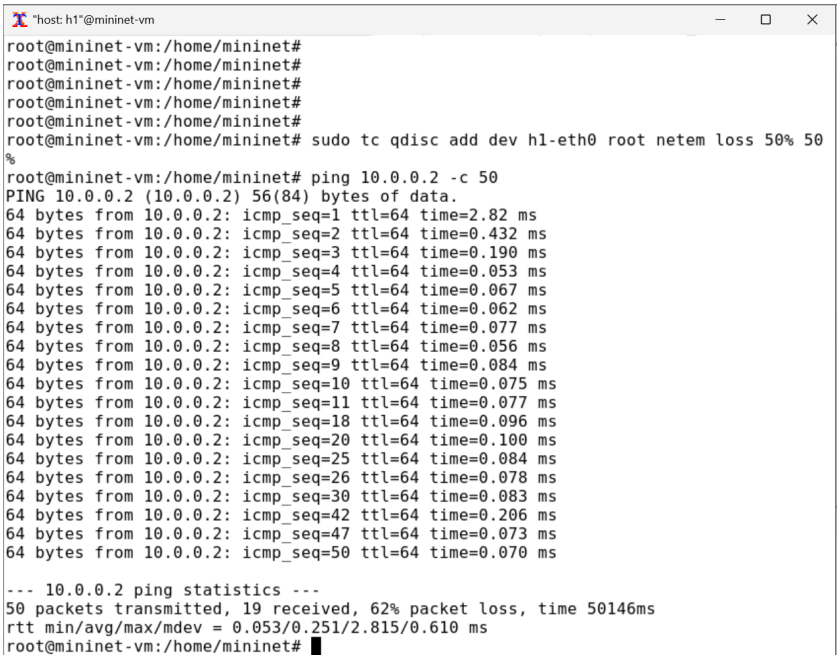

Рис. 4.8: Добавление значения корреляции для потери пакетов и тест

Восстановим конфигурацию интерфейса по умолчанию на узле h1. Добавим на интерфейсе узла h1 0,01% повреждения пакетов. Проверим конфигурацию с помощью инструмента iPerf3 для проверки повторных передач (рис. 4.9).

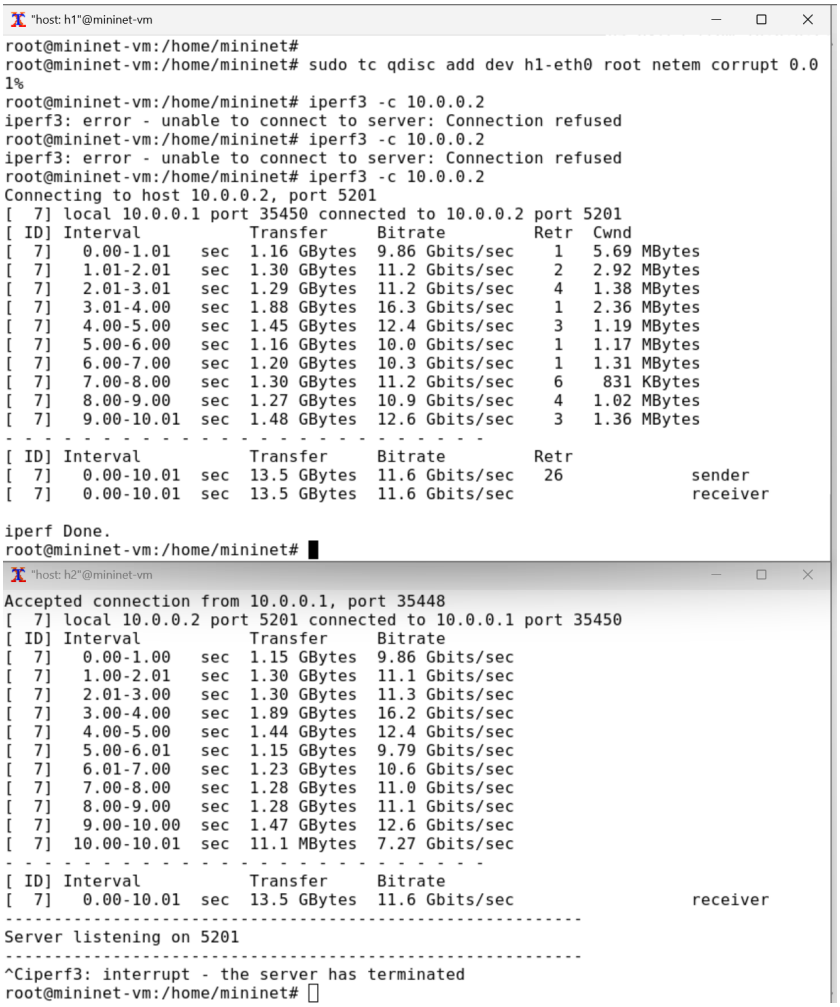

Рис. 4.9: Добавление на узле h1 0.01% повреждения пакетов и проврка через iperf

Восстановим конфигурацию интерфейса по умолчанию на узле h1. Добавим на интерфейсе узла h1 следующее правило: 25% пакетов (со значением корреляции 50%) будут отправлены немедленно, а остальные 75% будут задержаны на 10 мс. 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. 4.10).

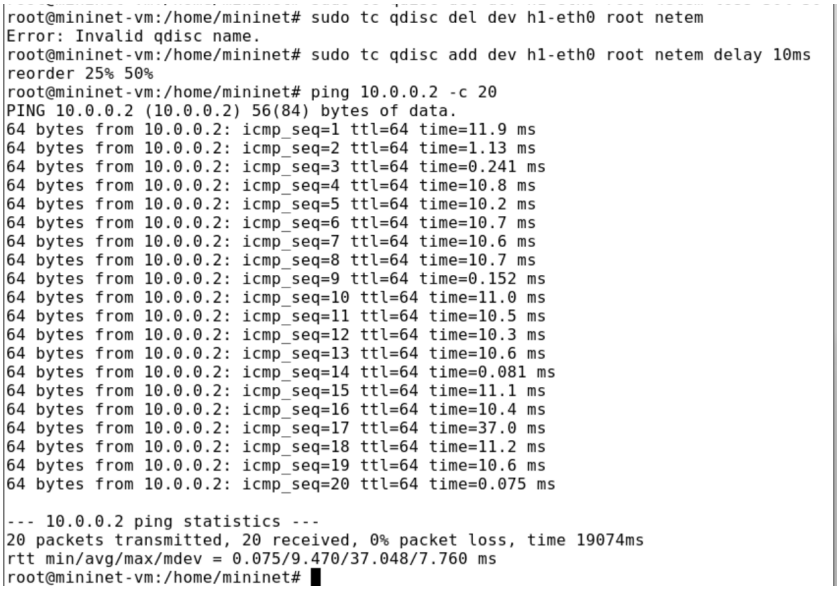

Рис. 4.10: Добавление переупорядочивания пакетов и тест

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): 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!(рис. 4.11).

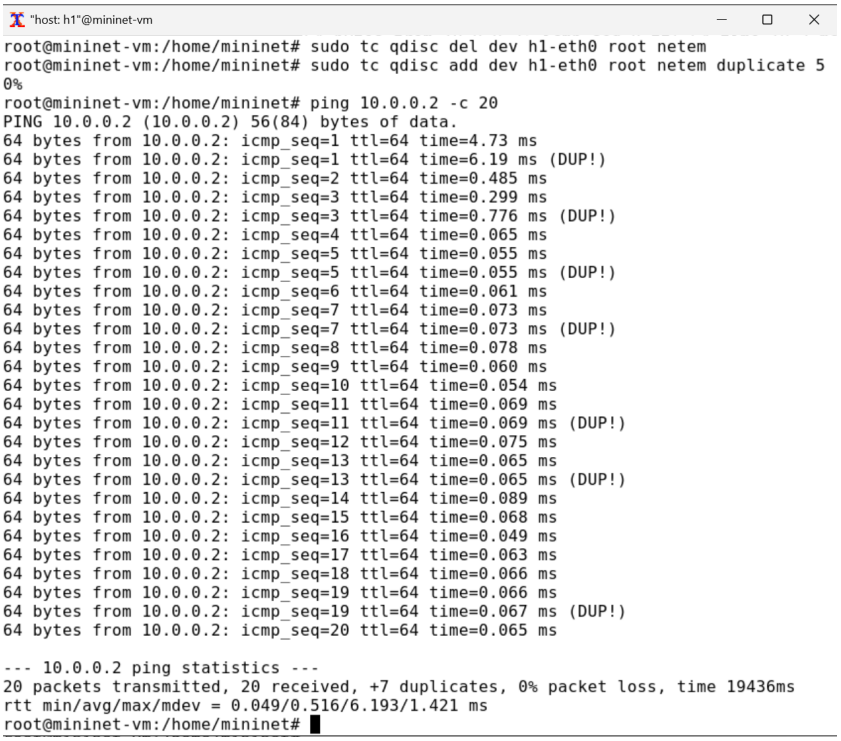

Рис. 4.11: Добавление дублирования пакетов и тест

Для каждого воспроизводимого эксперимента expname создадим свой каталог, в котором будут размещаться файлы эксперимента. В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-drop и перейдём в него. Создадим скрипт для эксперимента lab_netem_ii.py и внесём в него листинг программы(рис. 4.12).

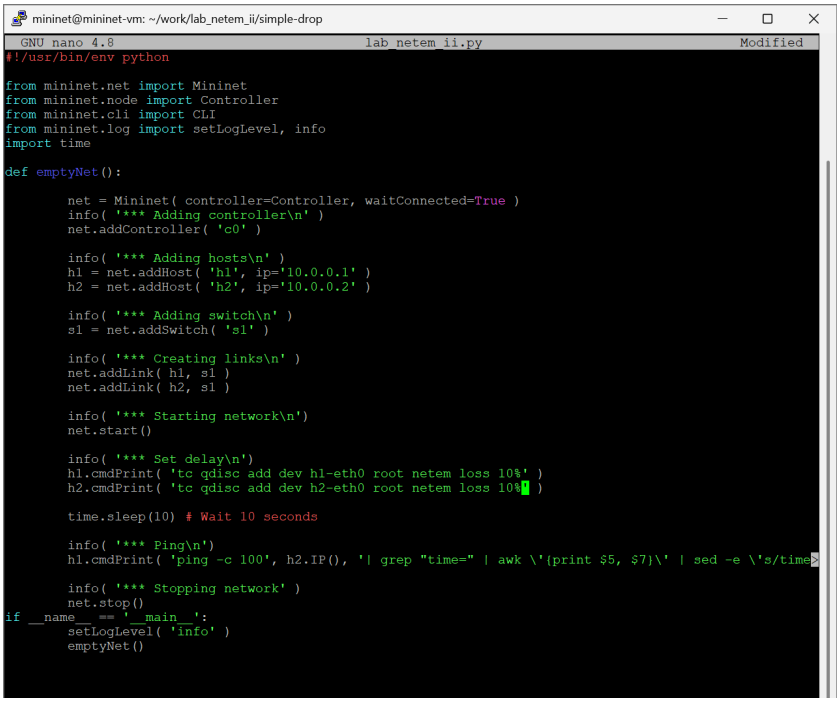

Рис. 4.12: Листинг lab_netem_ii для simple-drop из лабораторно

Скорректируем скрипт так, чтобы на экран или в отдельный файл выводилась информация о потерях пакетов(рис. 4.13).

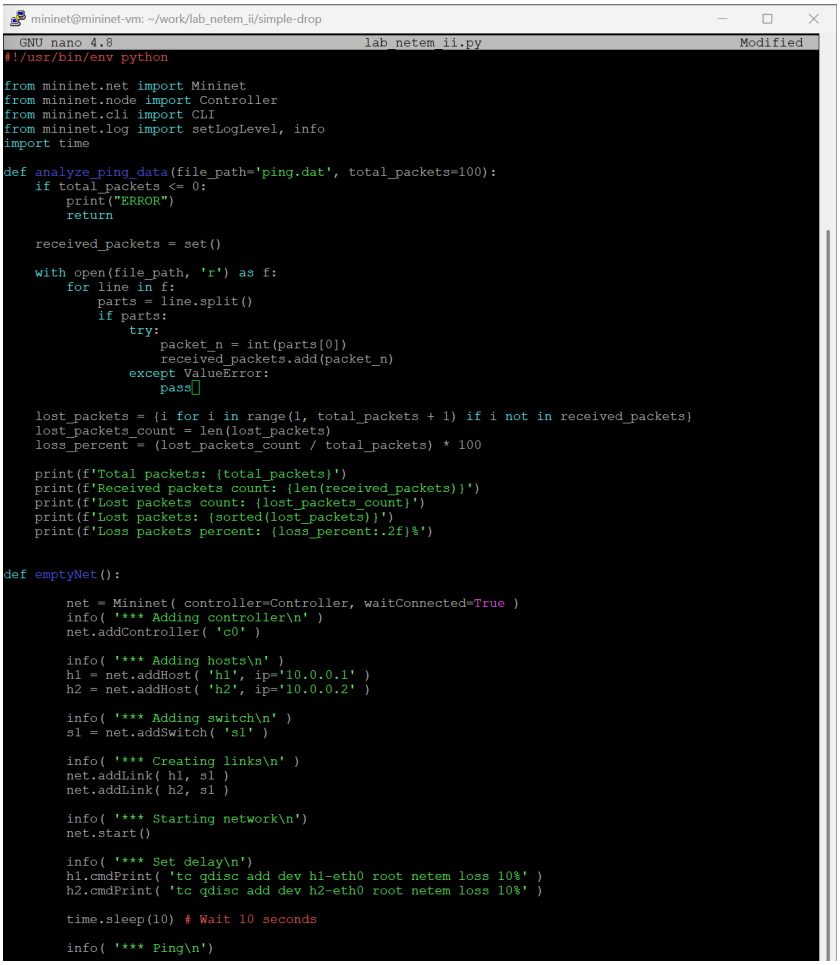

Рис. 4.13: Листинг lab_netem_ii для simple-drop с анализом

Создадим Makefile для управления процессом проведения эксперимента(рис. 4.14).

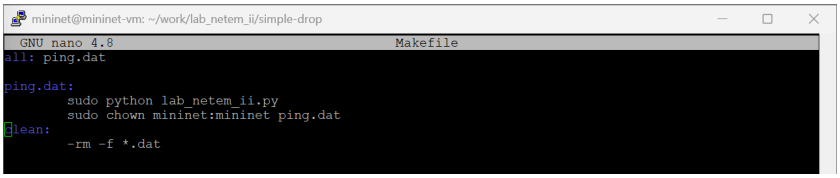

Рис. 4.14: Листинг Makefile для simple-drop

Выполним эксперимент и далее очистим каталог от результатов проведения экспериментов(рис. 4.15).

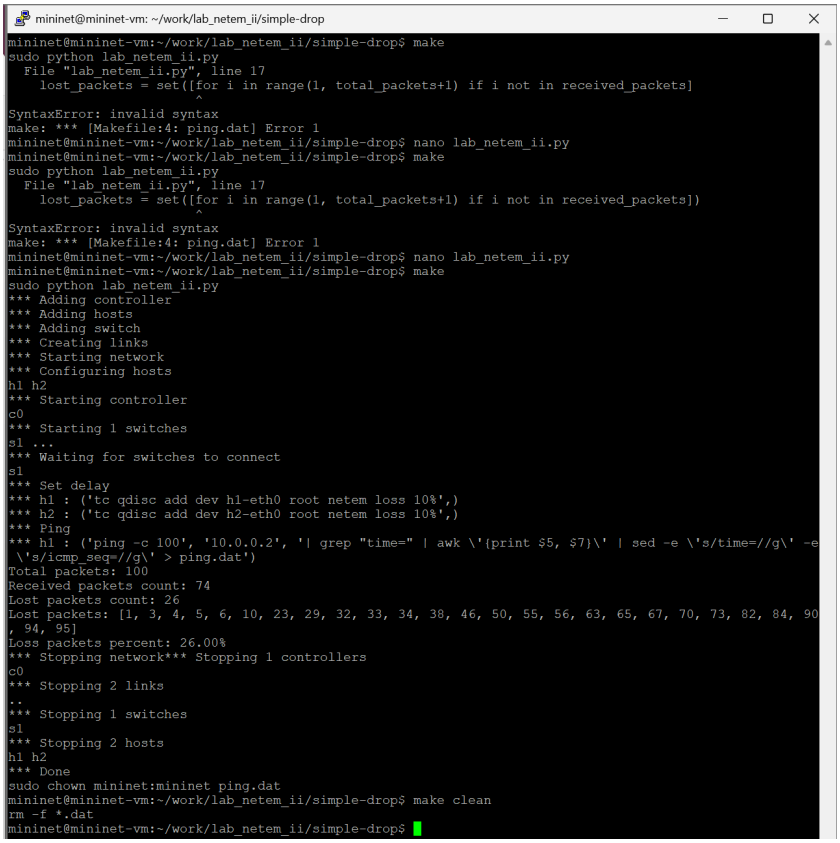

Рис. 4.15: Выполнение эксперимента и последующая очистка каталога

Теперь создадим каталоги основываясь на simple-drop под выполнение самостоятельной работы(рис. 4.16).

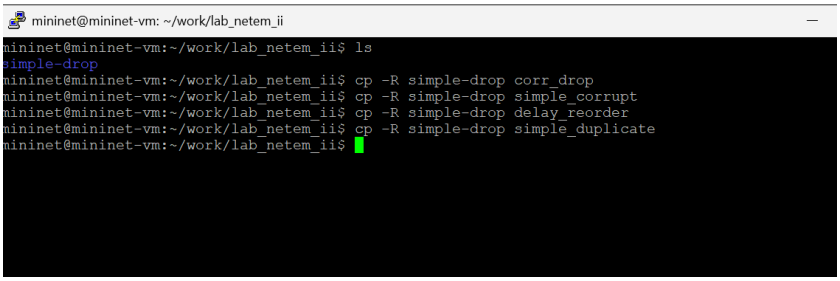

Рис. 4.16: Создание каталогов для самостоятельной работы

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

Скорректируем lab_netem_ii.py для эксепиремента по добавлению потери и коэффицента корреляции.(рис. 4.17)

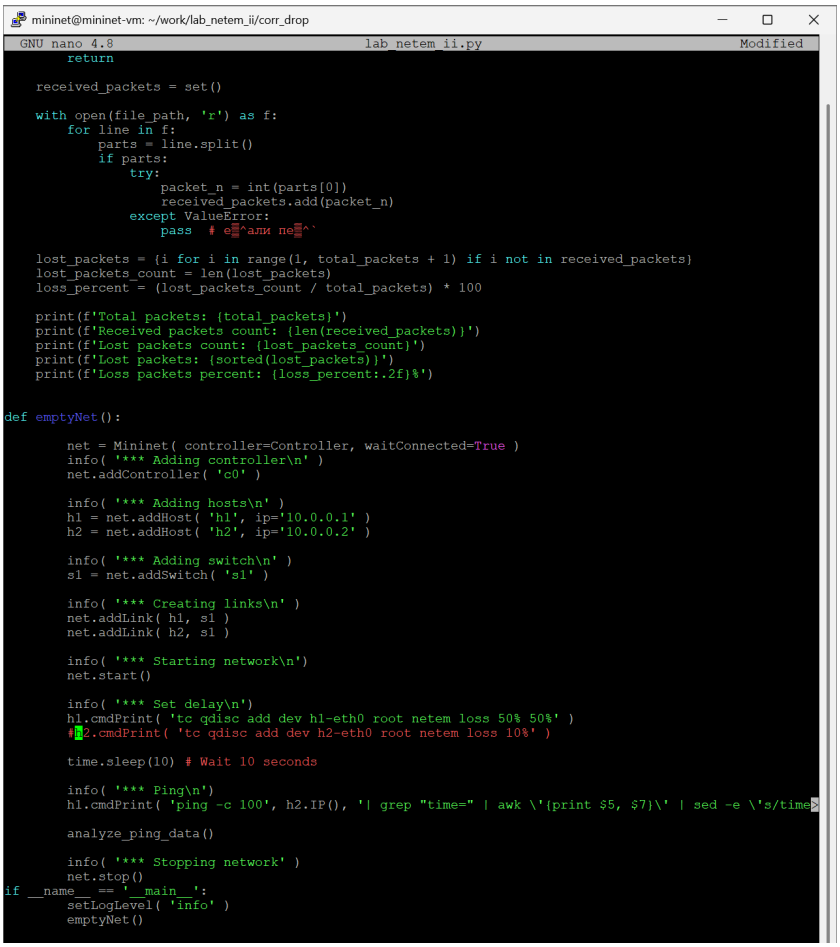

Рис. 4.17: Листинг программы для эксепиремента по добавлению потери и коэффицента корреляции

Запустим скрипт.(рис. 4.18)

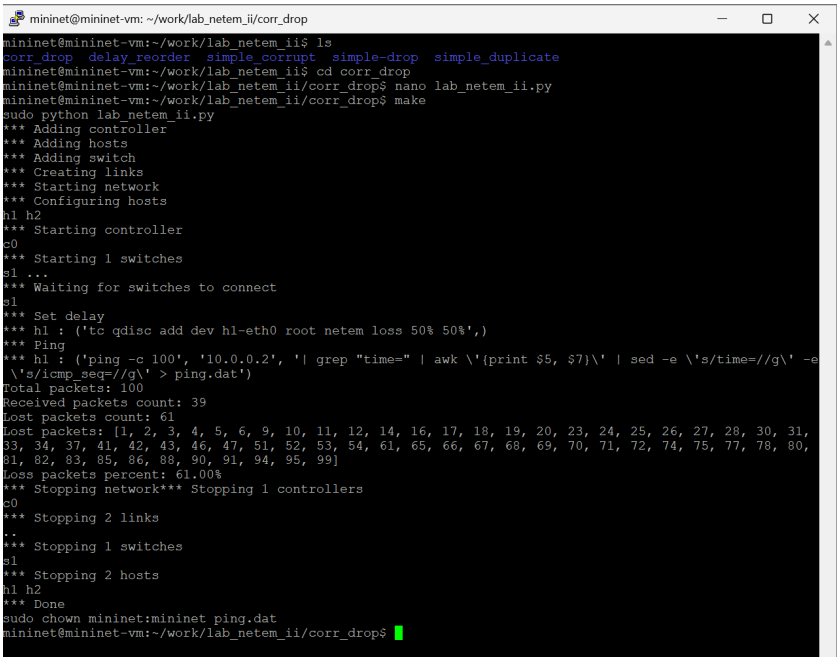

Рис. 4.18: Выполенение эксепиремента по добавлению потери и коэффицента корреляции

Скорректтируем lab_netem_ii.py для эксепиремента по повреждению пакетов.(рис. 4.19)

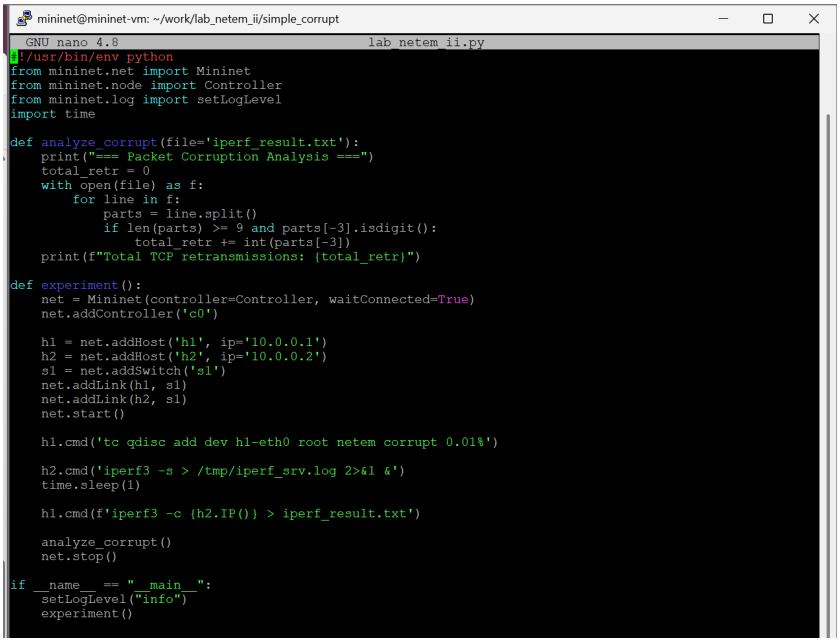

Рис. 4.19: Листинг программы для эксепиремента по повреждению пакетов

Запустим скрипт.(рис. 4.20)

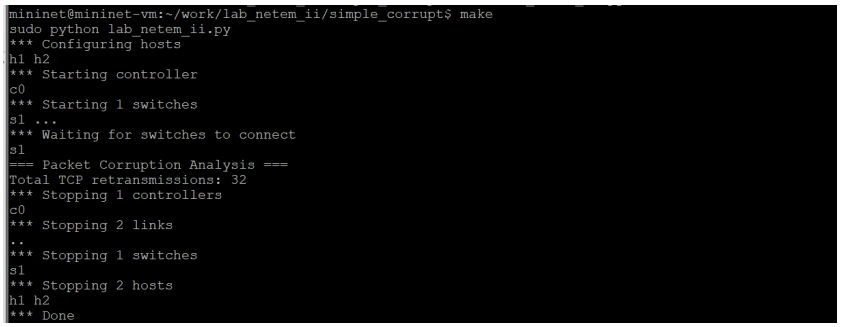

Рис. 4.20: Выполенение эксепиремента по повреждению пакетов

Скорретктируем lab_netem_ii.py для эксепиремента по изменению порядка пакетов.(рис. 4.21)

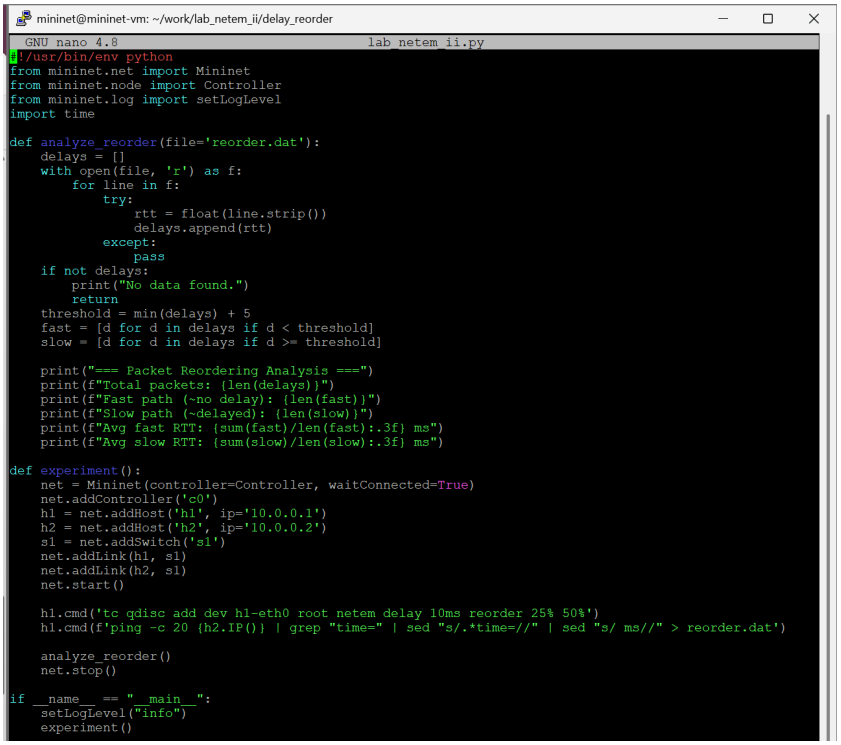

Рис. 4.21: Листинг программы для эксепиремента по изменению порядка пакетов

Запустим скрипт.(рис. 4.22)

```
mininet@mininet-vm:~/work/lab_netem_ii/delay_reorder$ nano lab_netem_ii.py
mininet@mininet-vm:~/work/lab_netem_ii/delay_reorder$ make
sudo python lab_netem_ii.py
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
=== Packet Reordering Analysis ===
Total packets: 20
Fast path (~no delay): 2
Slow path (~delayed): 18
Avg fast RTT: 0.060 ms
Avg slow RTT: 11.339 ms
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet reordero.dat
```

Рис. 4.22: Выполенение эксепиремента по изменению порядка пакетов

Скорретктируем lab_netem_ii.py для эксепиремента по дублированию пакетов.(рис. 4.23)

mininet@mininet-vm: ~/work/lab_netem_ii/simple_duplicate

GNU nano 4.8 lab_netem_ii.py

```
#!/usr/bin/env python
from mininet.net import Mininet
from mininet.node import Controller
from mininet.log import setLogLevel

def analyze_dup(file='dup.txt'):
    with open(file) as f:
        lines = f.readlines()
    duplicates = [l for l in lines if "DUP!" in l]
    print("=== Packet Duplication Analysis ===")
    print(f"Total packets: {len(lines)}")
    print(f"Duplicated packets: {len(duplicates)}")
    print(f"Duplication percent: {(len(duplicates)/len(lines))*100:.2f}%")

def experiment():
    net = Mininet(controller=Controller, waitConnected=True)
    net.addController('c0')
    h1 = net.addHost('h1', ip='10.0.0.1')
    h2 = net.addHost('h2', ip='10.0.0.2')
    s1 = net.addSwitch('s1')
    net.addLink(h1, s1)
    net.addLink(h2, s1)
    net.start()

    h1.cmd('tc qdisc add dev h1-eth0 root netem duplicate 50%')
    h1.cmd(f'ping -c 30 {h2.IP()} > dup.txt')

    analyze_dup()
    net.stop()

if __name__ == "__main__":
    setLogLevel("info")
    experiment()
```

Рис. 4.23: Листинг программы для эксепиремента по дублированию пакетов

Запустим скрипт.(рис. 4.24)

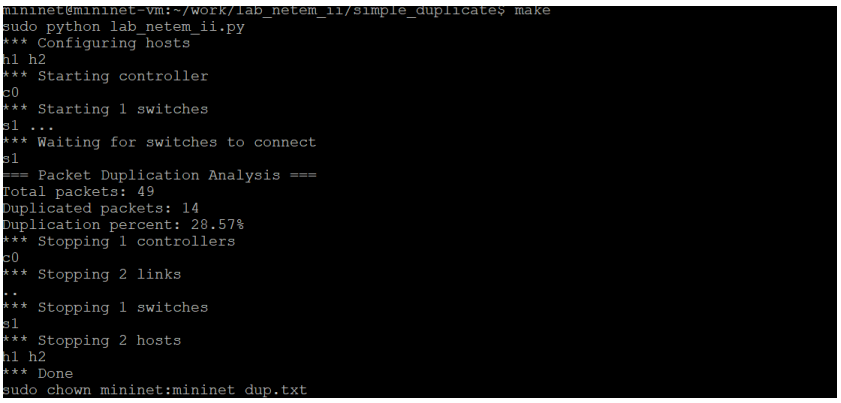

Рис. 4.24: Выполенение эксепиремента по дублированию пакетов

5 Выводы

В результате выполнения данной лабораторной работы я получил 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

1. Mininet [Электронный ресурс]. Mininet Project Contributors. URL: http://mininet.org/ (дата обращения: 07.10.2025).