

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Российский университет дружбы народов имени Патриса
Лумумбы»**

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

Направление: 09.03.03 Прикладная информатика

ОТЧЕТ

о прохождении учебной практики

Научно-исследовательская работа

(получение первичных навыков научно-исследовательской работы)

Место прохождения практики: отдел технической поддержки пользователей
(департамент технологических и информационных ресурсов) РУДН и
научные центры института прикладной математики и телекоммуникаций

ФИО: Саргсян Арам Грачьевич

Курс, группа 4, НПИбд-02-20

Студенческий билет № 1032201740

Руководители практики:

от РУДН к.ф.-м.н. Е.Г. Медведева

Научный руководитель:

к.ф.-м.н., доц. А.В. Королькова

Руководитель от организации:

д.т.н., проф. К.Е. Самуйлов

Оценка _____

г. Москва

2023 г.

Оглавление

Введение	3
1 Методы и материалы	5
1.1 NS-2	5
1.2 Mininet	5
1.2.1 Iperf3	6
1.2.2 Netem	6
1.3 Cisco Packet Tracer	7
1.4 GNS-3	8
2 Результаты	9
Заключение	13
Список литературы	15
Приложения	16

Введение

Согласно программе учебной практики по направлению 09.03.03 «Прикладная информатика» целями практики являются:

- формирование навыков использования современных научных методов для решения научных и практических задач;
- формирование универсальных, общепрофессиональных и профессиональных компетенций в соответствии с ОС ВО РУДН;
- формирование навыков проведения исследовательской работы;
- формирование навыков работы с источниками данных;
- знакомство с принципами функционирования и изучение методов разработки и анализа моделей функционирования сложных систем, их фрагментов и отдельных элементов;
- применение методов для анализа и расчёта показателей функционирования сложных систем, их фрагментов и отдельных элементов.

Также определены задачи практики:

- изучение специфики функционирования и соответствующих методов анализа сложных систем;
- формирование навыков решения конкретных научно-практических задач самостоятельно или в научном коллективе;
- формирование навыков проведения исследовательской работы и получении научных и прикладных результатов;
- изучение принципов и методов построения моделей сложных систем (в том числе технических систем, сетей и систем телекоммуникаций);

- изучение принципов и методов анализа поведения параметров моделей сложных систем (в том числе программных и технических систем, сетей и систем телекоммуникаций, и т.п.);
- приобретение практических навыков в области изучения научной литературы и (или) научно-исследовательских проектов в соответствии с будущим профилем профессиональной области.

Для достижения вышеупомянутых целей и задач в рамках учебной практики по теме «Моделирования алгоритма управления очередями RED в среде моделирования Mininet» мною было выполнено следующее:

- рассмотрены основные методы имитационного, аналитического и натурального моделирования сетей;
- исследована специфика моделирования различных сетей с помощью программы Mininet;
- проведен сравнительный анализ результатов натурального моделирования сети (построены и проанализированы графики размера TCP-окна, длины очереди и средней взвешенной длины очереди) при различных модификациях алгоритма RED, разных пороговых значений и типов TCP.

1 Методы и материалы

В этом разделе представим краткий обзор средств моделирования сетей передачи данных.

1.1 NS-2

NS-2 (Network simulator 2) — это программное средство моделирования сетей, использующееся для исследования и анализа поведения компьютерных сетей. Запуск имитационной модели в данной среде позволяет анализировать различные протоколы и алгоритмы сетевой связи.

NS-2 разработан на языке программирования C++ и TCL, что обеспечивает гибкость и расширяемость средства моделирования. NS-2 содержит библиотеку классов, которые представляют различные элементы сети, такие как узлы, маршрутизаторы, каналы связи и протоколы передачи данных. Для создания модели сети определяются характеристики и параметры каждого элемента сети: пропускная способность канала, задержки, вероятность потери пакетов и другие. После завершения симуляции NS-2 предоставляет мощные инструменты анализа результатов, включая возможность визуализации данных посредством программы NAM (Network animator), статистический анализ и сравнение результатов экспериментов, что позволяет изучать и оценивать производительность различных протоколов и алгоритмов в различных сценариях сети.

1.2 Mininet

Mininet — это симулятор сетевых топологий на основе виртуализации, который позволяет моделировать и изучать поведение сетей в контролируемой среде, основанный на использовании виртуальных машин и пространств имен Linux для создания изолированных сетевых узлов. Моделирование сетевых

топологий с помощью Mininet позволяет исследовать различные сетевые протоколы, маршрутизацию, управление трафиком и т.д. Возможности моделирования с помощью Mininet включают создание виртуальных сетевых узлов, конфигурирование топологий (связь между узлами, настраивать IP-адреса, маршрутизацию), имитировать различные условия сети, такие как задержки, потери пакетов и пропускную способность, интеграция с контроллерами для исследования новых протоколов и алгоритмов.

1.2.1 Iperf3

iPerf3 представляет собой кроссплатформенное клиент-серверное приложение с открытым исходным кодом, которое можно использовать для измерения пропускной способности между двумя конечными устройствами. iPerf3 может работать с транспортными протоколами TCP, UDP и SCTP:

TCP и SCTP:

- измерение пропускной способности
- возможность задать размер MSS/MTU
- отслеживание размера окна перегрузки TCP (CWnd)

UDP:

- измерение пропускной способности
- измерение потери пакетов
- измерение колебания задержки (jitter)
- поддержка групповой рассылки пакетов (multicast).

1.2.2 Netem

NETEM — это сетевой эмулятор Linux, используемый для тестирования производительности реальных клиент-серверных приложений в виртуальной сети. Модуль управляется при помощи команды `tc` из пакета `iproute2`. NETEM позволяет пользователю задать ряд параметров сети, такие как задержка, дрожание задержки (jitter), уровень потери пакетов, дублирование и изменение порядка пакетов. Данный эмулятор состоит из двух частей: модуля

ядра для организации очередей и утилиты командной строки для его настройки. Между протоколом IP и сетевым устройством создаётся очередь с дисциплиной обслуживания. Дисциплина обслуживания очереди реализуется как объект с двумя интерфейсами. Один интерфейс ставит пакеты в очередь для отправки, а другой интерфейс отправляет пакеты на сетевое устройство. На основе дисциплины обслуживания очередей принимается решение о том, какие пакеты отправлять, какие пакеты задерживать и какие пакеты отбрасывать. Дисциплины обработки очередей можно разделить на бесклассовые и классовые. Бесклассовые дисциплины, используемые по умолчанию в общем, получают данные, переупорядочивают, вносят задержку или уничтожают их. Наиболее распространённой бесклассовой дисциплиной является FIFO (первым пришёл, первым обслужен). Классовые дисциплины широко используются в случаях, когда тот или иной вид трафика необходимо обрабатывать по-разному. Примером классовой дисциплины может служить CBQ — Class Based Queueing (дисциплина обработки очередей на основе классов). Классы трафика организованы в дерево — у каждого класса есть не более одного родителя; класс может иметь множество потомков. Классы, которые не имеют родителей, называются корневыми. Классы, которые не имеют потомков, называются классами-ветками. Модуль управляется при помощи команды `tc` из пакета `iproute2`.

1.3 Cisco Packet Tracer

Packet Tracer — это программное средство, предоставляемое компанией Cisco Systems, позволяющей смоделировать, конфигурировать и отлаживать сетевые сценарии, широко используемое в области сетевых технологий. Данное программное обеспечение предоставляет виртуальную среду, которое позволяет создавать сетевые топологии и настраивать устройства Cisco: маршрутизаторы, коммутаторы и т.д. Графический интерфейс позволяет соединять устройства, устанавливать параметры соединений и задавать настройки протоколов. Cisco Packet Tracer позволяет имитировать передачу данных в сети. Пользователи могут выполнять различные тесты связи, проводить диагностику и мониторинг сетевых устройств, а также создавать и анализировать журналы событий.

1.4 GNS-3

GNS-3 — это программное средство моделирования сетей, позволяющий создавать виртуальные сети, состоящие из реальных или виртуальных устройств, и анализировать их поведение. GNS-3 разработан на языке программирования Python и основан на эмуляторе динамических узлов Dynamips, который позволяет запускать реальные образы операционных систем. В отличие от Packet Tracer, GNS-3 позволяет смоделировать не только устройства Cisco, но и другие устройства, например, Juniper, Palo, Alto и другие, что позволяет смоделировать различные типы сетей, включая центры обработки данных и облачные инфраструктуры. Одной из главных особенностей GNS-3 является интеграция с виртуальными машинами, что расширяет возможности моделирования. Появляется возможность создавать сетевые сценарии, в которых виртуальные машины выполняют реальные функции, такие как серверы, клиенты, точки доступа Wi-Fi и т.д. Это позволяет проводить натурное моделирование и получить более реалистичные результаты в рамках виртуальной среды.

2 Результаты

Мною была написана программа, реализующую натурную модель с базовой топологией с двумя хостами и одним коммутатором для реализации программы.

Полная реализация программы приведена в разделе **Приложения**, для вывода графиков была использована программа GNUPLOT.

Смоделировал сеть три раза с указанными параметрами. В первом случае добавили задержку в 100 мс на обоих хостах, во втором случае 50мс, в третьем случае 100мс с отклонением в 10мс и запустив gnuplot-скрипт, я получил следующий график(рис. 2.1).

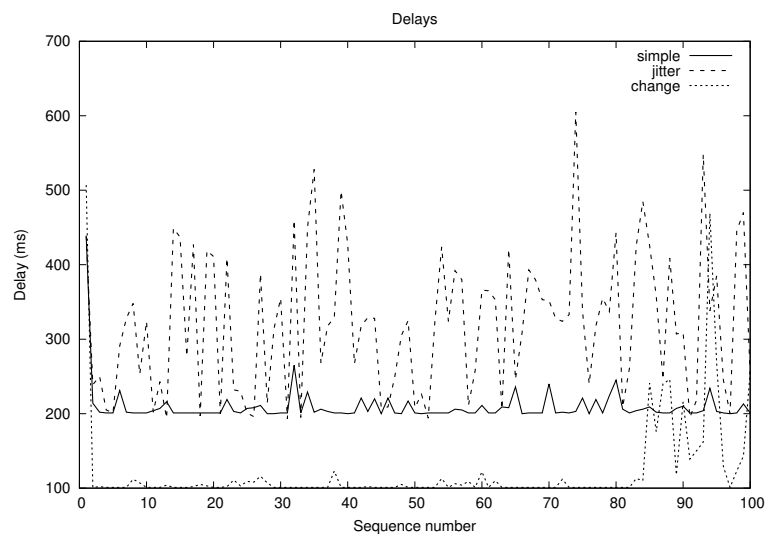


Рис. 2.1: Измерение задержек при разных условиях

Как мы видим из графика, наименьшие задержки имеет сеть с задержкой в 50 мс, при указании отклонения в 10мс задержки возрастают в несколько раз.

На коммутаторе s1 указал дисциплину очереди red со следующими параметрами: минимальный порог сброса в 30000 байтов, максимальный в 60000 байтов. Смоделировал сеть и с помощью iperf3 получил графики окна перегрузки, пропускной способности и количества переданных байтов. Изменил

дисциплину очереди с red на ared. Вывел соответствующие графики также и для этой модификации (рис. 2.2, 2.3, 2.4, 2.5, 2.6, 2.7).

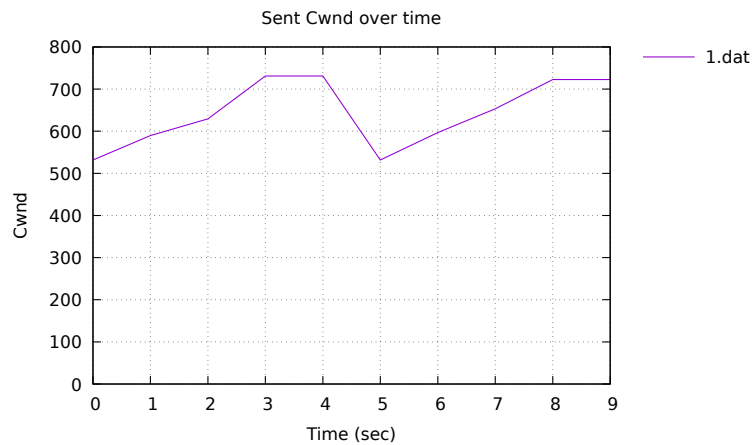


Рис. 2.2: Окно перегрузки при использовании red

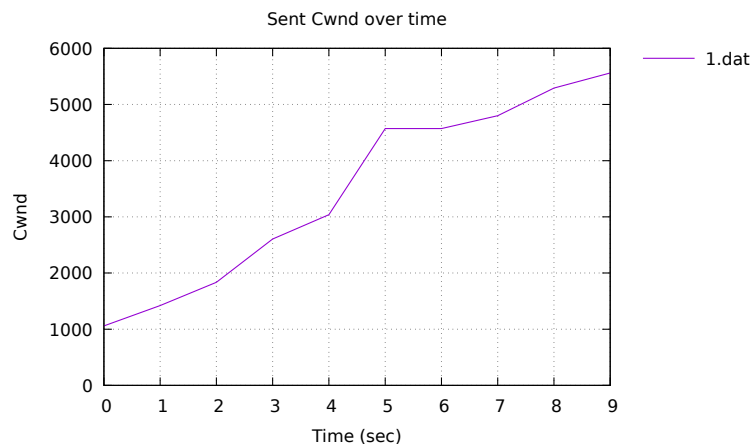


Рис. 2.3: Окно перегрузки при использовании ared

Как мы видим, при классическом алгоритме размер TCP окна перегрузки значительно меньше.

Сравнивая графики пропускной способности, мы видим, что adaptive red имеет в среднем немного большую пропускную способность.

При ARED также имеем большее количество переданных байтов.

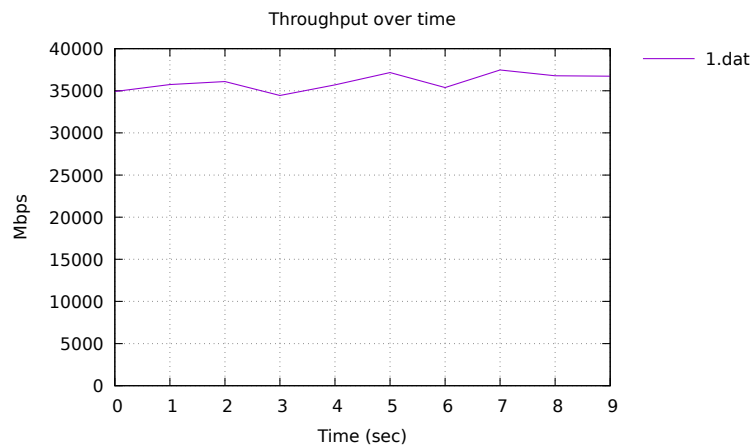


Рис. 2.4: Пропускная способность при использовании red

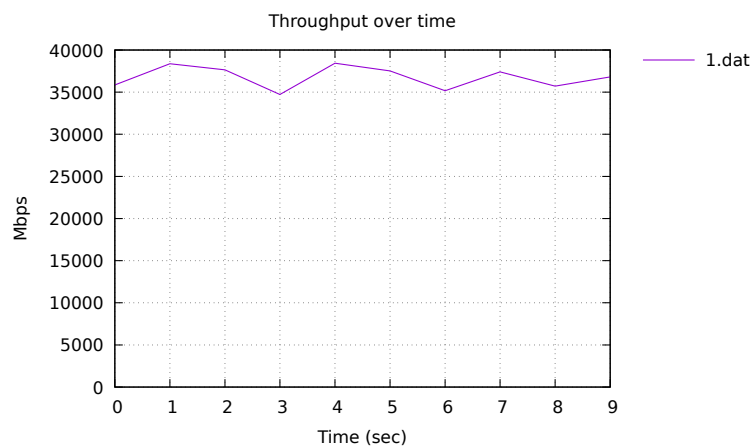


Рис. 2.5: Пропускная способность при использовании aged

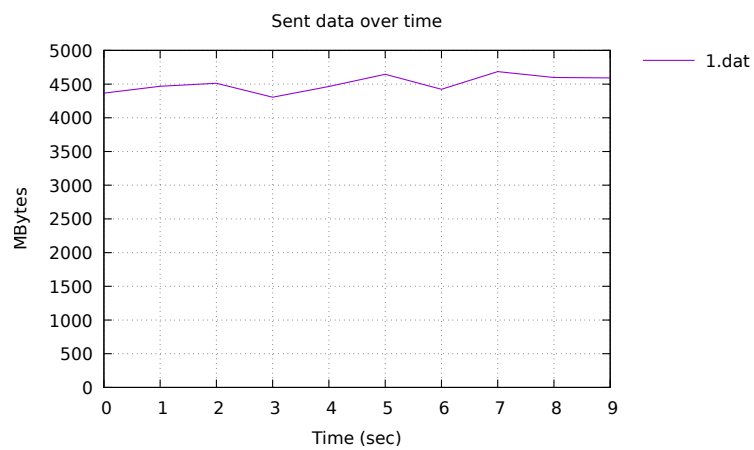


Рис. 2.6: Количество переданных данных при использовании red

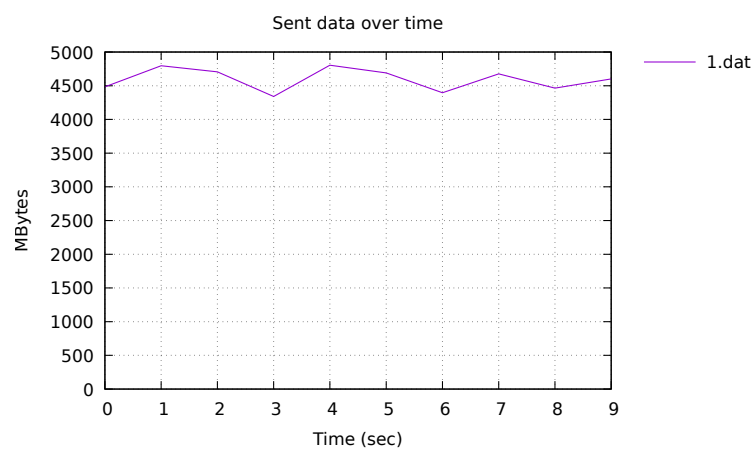


Рис. 2.7: Количество переданных данных при использовании ared

Заключение

За период практики в отделе технической поддержки пользователей (департамент технологических и информационных ресурсов) РУДН и научных центрах института прикладной математики и телекоммуникаций. были достигнуты все цели и решены все задачи, определенные в программе научной практики направления подготовки 09.03.03 «Прикладная информатика» программы «Прикладная информатика» (см. введение отчёта по практике). В процессе прохождения практики я работал с научной терминологией области исследований; научился собирать и обрабатывать данные, необходимые для формирования соответствующих выводов исследований; осуществлять целенаправленный поиск информации на русском и английском языках о новейших научных достижениях в Интернете и из других источников; строить и анализировать имитационные и натурные модели объекта исследований.

В результате прохождения данной практики я приобрел следующие практические навыки, умения, универсальные и профессиональные компетенции:

- способность управлять проектом на всех этапах его жизненного цикла (постановка задачи, планирование, реализация);
- способность составлять естественно-научные отчеты с IMRAD структурой (введение, методы и материалы, результаты и дискуссия);
- способность разрабатывать имитационные модели и проводить их анализ при решении задач в профессиональной области (составлена имитационная модель сети с алгоритмом управления очередью на маршрутизаторе типа RED);
- способность проведения работ по обработке и анализу научно-технической информации и результатов исследований (изучение необходимой литературы по теме исследования на русском и английском языках, подготовка литературного обзора по теме исследований).

Таким образом, в рамках практики я рассмотрел моделирование модуля RED с помощью программного средства Mininet. Также представлена программная реализация натурной модели сети модулем RED и проведен сравнительный анализ результатов при моделировании сети с разными входными параметрами, модификаций RED и типов TCP.

Список литературы

Приложения

Ссылка на репозиторий: https://github.com/agsargsyan/study_2022-2023_practice

Программа симуляции топологии с red

```
1  -#!/usr/bin/env python
2
3  """
4  Simple experiment.
5  Output: ping.dat
6  """
7  from mininet.net import Mininet
8  from mininet.node import Controller
9  from mininet.cli import CLI
10
11 from mininet.log import setLogLevel, info
12 import time
13
14 def emptyNet():
15
16     "Create an empty network and add nodes to it."
17
18     net = Mininet( controller=Controller, waitConnected=True )
19
20     info( '*** Adding controller\n' )
21     net.addController( 'c0' )
22
23     info( '*** Adding hosts\n' )
```



```

24 h1 = net.addHost( 'h1', ip='10.0.0.1' )
25 h2 = net.addHost( 'h2', ip='10.0.0.2' )
26
27 info( '*** Adding switch\n' )
28 s1 = net.addSwitch( 's1' )
29
30 info( '*** Creating links\n' )
31 net.addLink( h1, s1, bw = 10 )
32 net.addLink( h2, s1, bw = 10 )
33
34 info( '*** Starting network\n' )
35 net.start()
36
37 info( '*** Set red options\n' )
38 s1.cmdPrint( 'tc qdisc add dev s1-eth1 root handle 1: red limit
    ↪ 1000000 max 30000 min 60000 burst 80 avpkt 1000 bandwidth
    ↪ 10Mbit ' )
39
40 #для ARED
41 #s1.cmdPrint( 'tc qdisc add dev s1-eth1 root handle 1: red
    ↪ limit 1000000 max 30000 min 60000 burst 80 avpkt 1000
    ↪ bandwidth 10Mbit adaptive' )
42
43
44 info( '*** Traffic generation\n' )
45 h2.cmdPrint( 'iperf3 -s -D -1 ' )
46 time.sleep(21) # Wait 21 seconds
47 h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )
48
49 info( '*** Ping\n' )
50 h1.cmdPrint( 'ping -c 10', h2.IP(), '| grep "time=" | awk
    ↪ \' {print $5, $7} \\' | sed -e \'s/time=//g\'
    ↪ -e \'s/icmp_seq=//g\' > ping.dat' )
51
52 info( '*** Stopping network' )

```

```

53     net.stop()
54
55     if __name__ == '__main__':
56         setLogLevel( 'info' )
57         emptyNet()
58
59 % \end{verbatim}

```

Makefile

```

1 all: ping.dat ping.pdf plot
2
3 ping.dat:
4     sudo python lab_netem_i.py
5     sudo chown mininet:mininet ping.dat
6
7 ping.pdf: ping.dat
8     ./ping_plot
9
10 plot: iperf_result.json
11     plot_iperf.sh iperf_result.json
12
13 clean:
14     -rm -f *.dat *.pdf *.json *.csv
15     -rm -rf results

```

```

1
2 #!/usr/bin/gnuplot --persist
3
4 set terminal png crop
5 set output 'ping.pdf'
6 set xlabel "Sequence number"
7 set ylabel "Delay (ms)"

```

```
8 set grid
9 plot "ping.dat" with lines
10
```