

# Моделирование модуля активного управления трафиком сети передачи данных

Выполнил студент: С. М. Наливайко

Группа НФИбд-01-18  
Студенческий билет № 1032183644



Российский университет  
дружбы народов

Руководитель выпускной  
квалификационной работы:  
Королькова А. В., к.ф.-м.н.,  
доцент, доцент  
кафедры прикладной информатики  
и теории вероятностей

# Цель

Целью моей работы является моделирование поведения двух типов TCP-подобного трафика в средстве натурного моделирования Mininet.

# Задачи

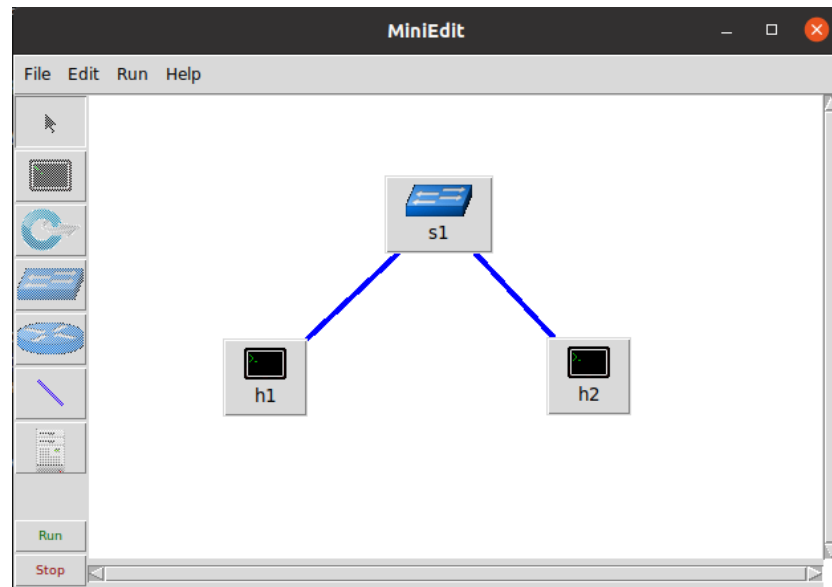
Основными задачами данной работы являются:

1. Построить модель активного управления трафиком в Mininet.
2. Измерение и анализ характеристик производительности моделируемой сети.
3. Автоматизация процесса моделирования и измерения сетевых характеристик в среде Mininet с помощью единого программного модуля.

# Mininet

```
sergey@sergey: ~  
sergey@sergey:~$ sudo mn  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet>
```

Утилита mn



Miniedit

# Mininet

```
1 from mininet.link import TCLink
2 from mininet.net import Mininet
3 from mininet.node import CPULimitedHost
4 from mininet.topo import Topo
5
6 '''
7 Создание простой топологии:
8 10 хостов, 1 коммутатор, 10 соединений
9 '''
10 class CustomTopology(Topo):
11     def __init__(self, **opts):
12         super(CustomTopology, self).__init__(**opts)
13         s1 = self.addSwitch(name="s1")
14         for i in range(1, 11):
15             host = self.addHost(name="h%d" % i, ip="10.0.0.%d" % i)
16             self.addLink(host, s1)
17
18 '''
19 Запуск сети и проверка достижимости элементов
20 '''
21 if __name__ == "__main__":
22     topology = CustomTopology()
23     net = Mininet(topo=topology, host=CPULimitedHost, link=TCLink)
24     net.start()
25     print("Сеть заработала")
26     net.pingAll()
27     net.stop()
28     print("Сеть остановилась")
```

Создание виртуальной сети с  
помощью библиотеки mininet

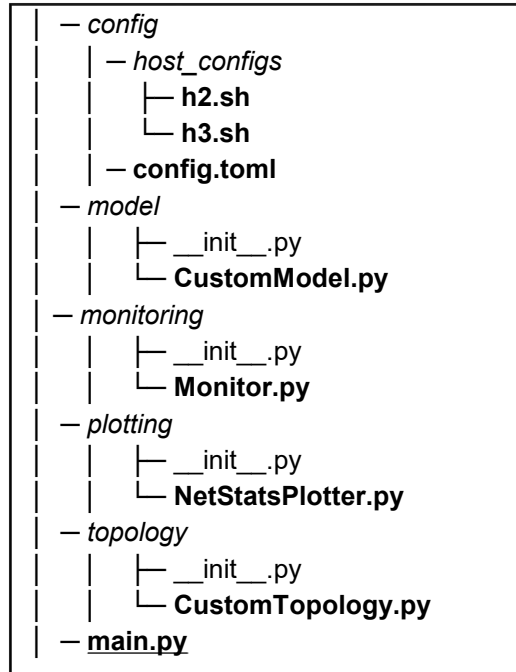


# Вспомогательные утилиты

- iproute2 - набор утилит для управления параметрами сетевых устройств в ядре Linux.
- iPerf3 - утилита генерации сетевого трафика.



# Структура приложения



# Шаблон конфигурационного файла

```
1  # device settings
2  [devices]
3      [devices.h1]
4          name = "h1"
5          ip = "10.0.0.1"
6          cmd = [
7
8          ]
9      [devices.h2]
10         name = "h1"
11         ip = "10.0.0.2"
12         cmd = [
13
14         ]
15
16 # switch settings
17 [switches]
18     [switches.s1]
19         name = "s1"
```

Шаблон конфигурационного файла



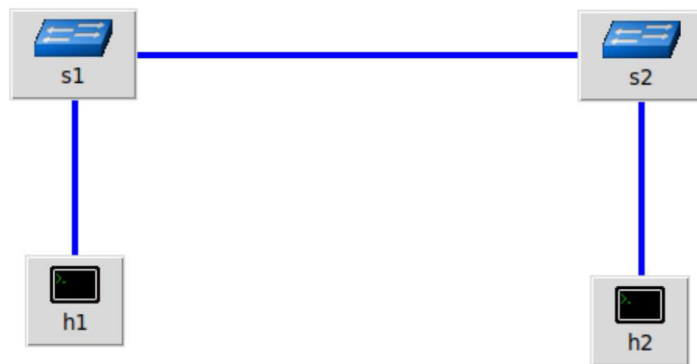
# Шаблон конфигурационного файла

```
21  # link settings
22  [links]
23  pairs = [
24      ["h1", "s1"],
25      ["s1", "h2"]
26  ]
27  cmd = [
28  ]
29
30  [monitoring]
31  monitoring_time = 60
32  monitoring_interval = 0.1
33  host_client = "h1"
34  host_server = "h2"
35  interface = "s1-eth1"
36  iperf_file_name = "iperf.json"
37  iperf_flags = "-b 15mbit"
38  queue_data_file_name = "qlen.data"
39  plots_dir = "plots_dir_second"
```

Шаблон конфигурационного файла



# Топология 1



Топология 1

# Окно перегрузки

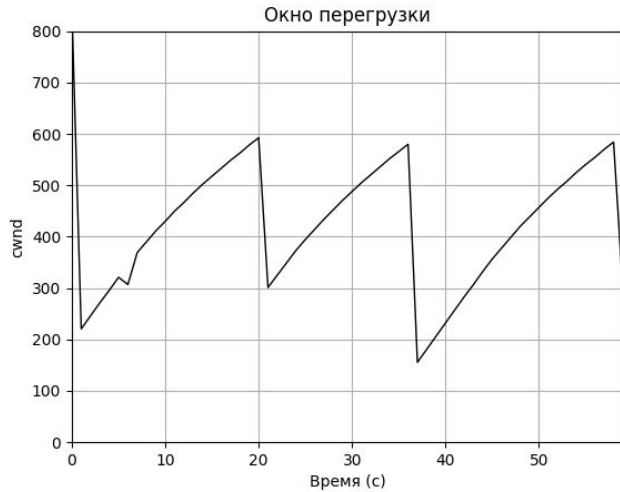


График изменения окна перегрузки для TCP Reno

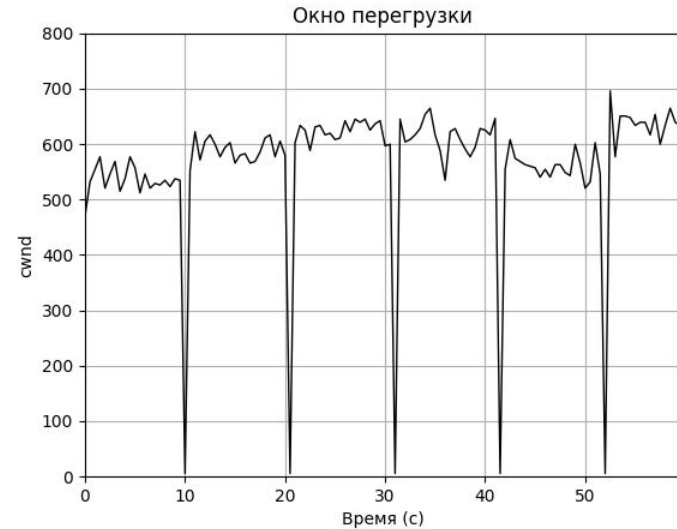


График изменения окна перегрузки для TCP BBR

# Длина очереди на интерфейсе s2-eth2

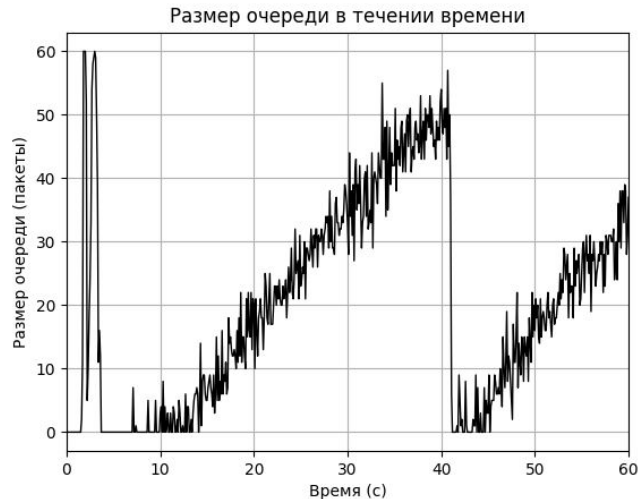


График изменения длины очереди для TCP Reno

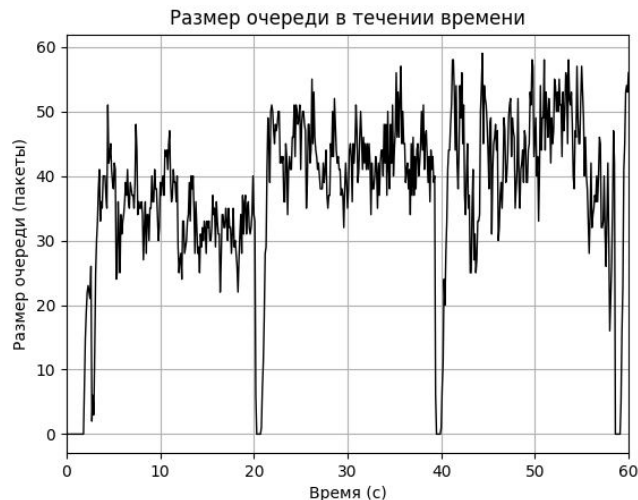


График изменения длины очереди для TCP BBR

# Пропускная способность

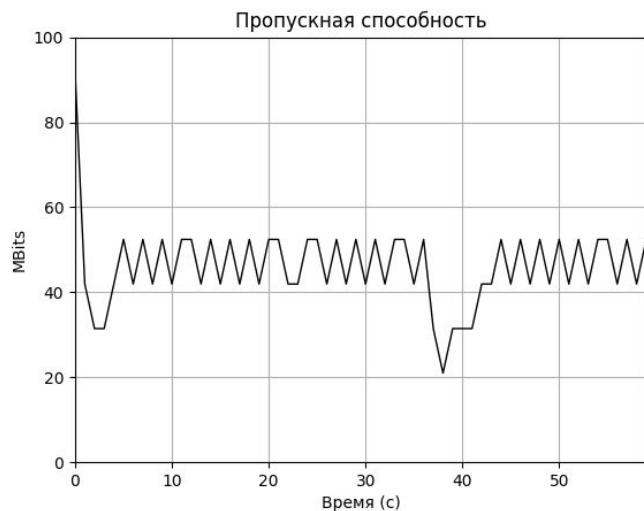


График изменения пропускной способности для TCP Reno

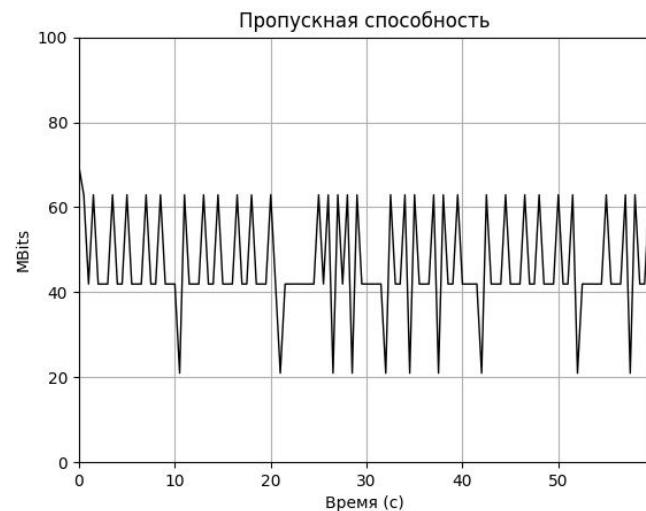
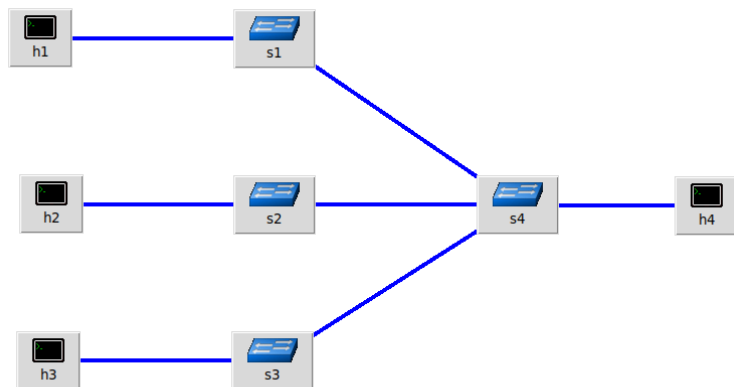


График изменения пропускной способности для TCP BBR

# Топология 2



Топология 2

# Окно перегрузки

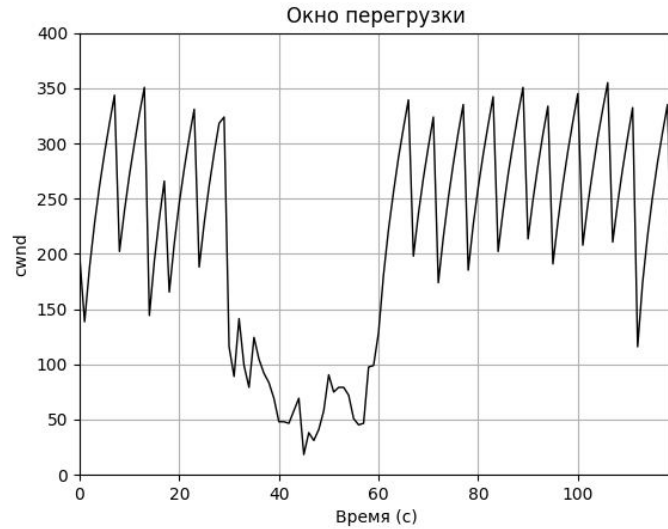


График изменения окна перегрузки для TCP Reno

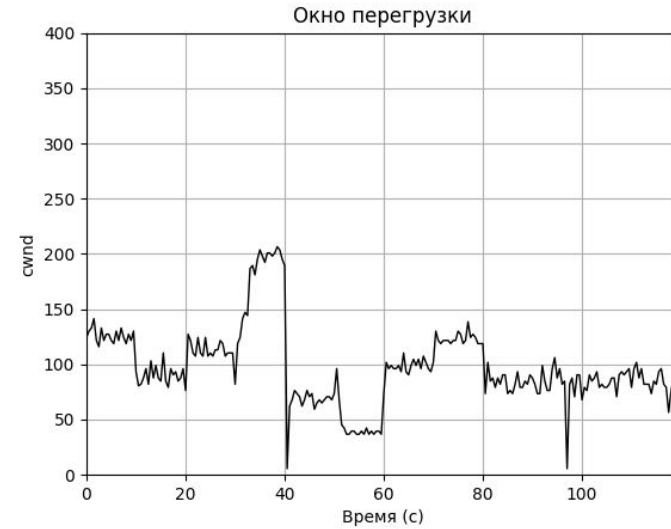


График изменения окна перегрузки для TCP BBR

# Длина очереди на интерфейсе s4-eth4

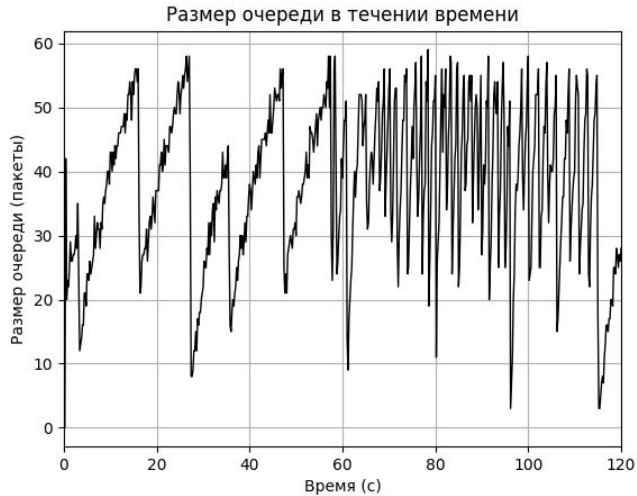


График изменения длины очереди для TCP Reno

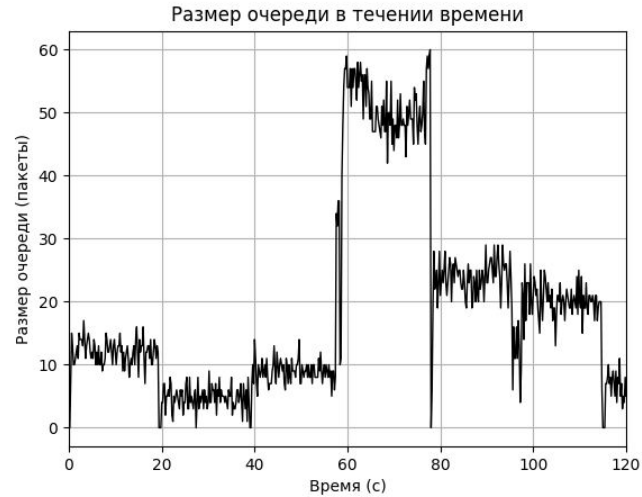


График изменения длины очереди для TCP BBR



# Пропускная способность

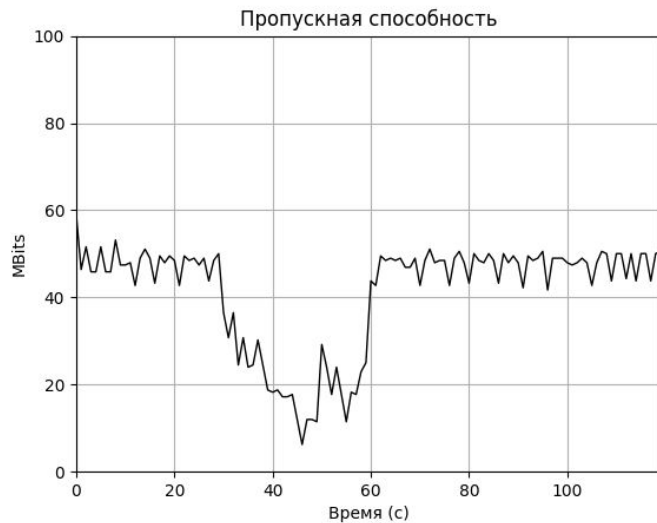


График изменения пропускной способности для TCP Reno

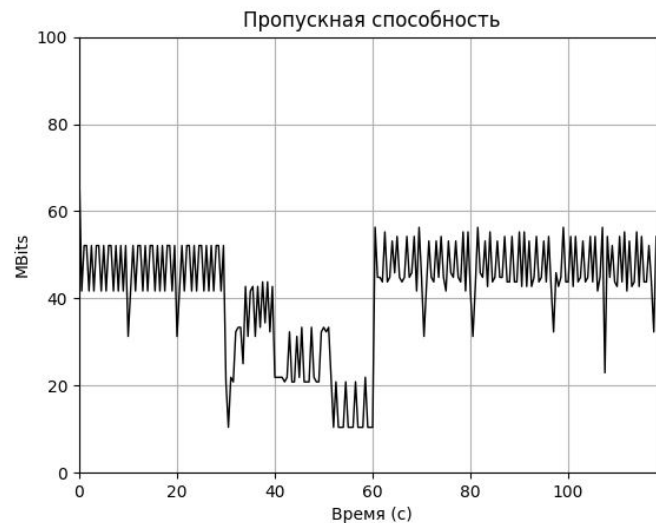


График изменения пропускной способности для TCP BBR

# Список литературы

1. Implementation of simplified custom topology framework in Mininet / Chandan Pal S Veena, Ram P. Rustagi, K. N. B. Murthy // 2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE).—2014.
2. Performance Analysis of Congestion Control Mechanism in Software Defined Network (SDN) / M. Z. A. Rahman, N. Yaakob, A. Amir et al.—2017.
3. tc-netem(8) — Linux manual page. — URL: <https://man7.org/linux/man-pages/man8/tc-netem.8.html>.
4. Metrics for the Evaluation of Congestion Control Mechanisms : RFC : 5166 / RFC Editor ; Executor: S. Floyd : 2008. — 03. — URL: <https://www.rfc-editor.org/rfc/rfc5166.txt>.
5. Mininet.—URL: <http://mininet.org/>.
6. Mininet Python API Reference Manual. — URL: <http://mininet.org/api/annotated.html>.
7. iPerf 3 user documentation.—URL: <https://iperf.fr/iperf-doc.php>.
8. tc(8) — Linux manual page. — URL: <https://man7.org/linux/man-pages/man8/tc.8.html>.
9. Наливайко С. М. Автоматизация процессов моделирования и измерения сетевых характеристик в Mininet // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем: материалы Всероссийской конференции с международным участием.—2022.—Р. 397-403.



# Заключение

В рамках данной работы:

1. Построили модель активного управления трафиком в Mininet.
2. Автоматизировали процесс моделирования и измерения сетевых характеристик с помощью программы, написанной на языке программирования высокого уровня python.
3. Промоделировали поведение двух типов ТСП-подобного трафика средствами созданной программы и рассмотрели их сетевые характеристики.