

Автоматизация процессов моделирования и измерения сетевых характеристик в Mininet

С. М. Наливайко

Information and Telecommunication Technologies and
Mathematical Modeling of High-Tech Systems 2022

ИТТММ 2022
18 - 22
3 Ordzhonikidze st.



Российский университет
дружбы народов

Цель

Целью данной работы является построение средства автоматизации процессов моделирования и измерения сетевых характеристик передачи данных. В качестве среды моделирования была выбрана программа Mininet, так как она позволяет использовать реальные сетевые приложения, сетевые протоколы и ядро Unix/Linux для тестирования и анализа характеристик, моделируемых в ней компьютерных сетей и сетевых протоколов, а в качестве программ, которые позволяют измерять сетевые характеристики, были выбраны iproute2 и iperf3.

Создание простой сети.

Характеристики

- скорость передачи данных ограничена;
- максимальная пропускная способность соединения $h1-s1$ равна 100 Мбит/с;
- максимальная пропускная способность соединения $s2-h2$ равна 50 Мбит/с;
- на коммутаторе $s2$ стоит дисциплина обработки очередей FIFO с максимальным количеством пакетов, равным 30;
- потери в сети составляют 0.001%;
- задержка имеет нормальное распределение с математическим ожиданием в 30 мс и с дисперсией в 7 мс.
- в сети работает алгоритм для работы с перегрузками TCP Reno.

Топология создаваемой сети

Пример простой топологии, состоящей из двух хостов h1 и h2, а также соединяющих их двух коммутаторов s1 и s2, представлен на рис. 1.



Рис. 1. Пример топологии

Автоматизация процесса создания сети

Метод автоматизации основан на создании конфигурационного toml-файла, из которого программа на языке программирования Python будет считывать данные и учитывать изменения при настройке параметров топологии, мониторинговых характеристик.



Автоматизация процесса создания сети. Конфигурационный файл

```
1  # device settings
2  [devices]
3      [devices.h1]
4          name = "h1"
5          ip = "10.0.0.1"
6          cmd = [
7              "sysctl -w net.ipv4.tcp_congestion_control=reno"
8          ]
9      [devices.h2]
10         name = "h2"
11         ip = "10.0.0.2"
12         cmd = [
13             "sysctl -w net.ipv4.tcp_congestion_control=reno"
14         ]
15
16 # switch settings
17 [switches]
18     [switches.s1]
19         name = "s1"
20     [switches.s2]
21         name = "s2"
```

Листинг 1.1. Конфигурационный файл

Автоматизация процесса создания сети. Конфигурационный файл

```
23 # link settings
24 [links]
25 pairs = [
26     ["h1", "s1"],
27     ["s1", "s2"],
28     ["s2", "h2"]
29 ]
30 cmd = [
31     "tc qdisc replace dev s1-eth2 root handle 10: tbf rate 100mbit burst 50000 limit 150000",
32     "tc qdisc add dev s1-eth2 parent 10: handle 20: netem loss 0.001% delay 30ms 7ms distribution normal",
33     "tc qdisc replace dev s1-eth1 root handle 10: tbf rate 100mbit burst 50000 limit 150000",
34     "tc qdisc add dev s1-eth1 parent 10: handle 20: netem loss 0.001% delay 30ms 7ms distribution normal",
35     "tc qdisc replace dev s2-eth2 root handle 10: tbf rate 50mbit burst 25000 limit 75000",
36     "tc qdisc add dev s2-eth2 parent 10: handle 15: pfifo limit 30",
37     "tc qdisc replace dev s2-eth1 root handle 10: tbf rate 50mbit burst 25000 limit 75000"
38 ]
```

Листинг 1.2. Конфигурационный файл

Автоматизация процесса мониторинга.

Конфигурационный файл

```
40  [monitoring]
41  monitoring_time = 240
42  monitoring_interval = 0.1
43  host_client = "h1"
44  host_server = "h2"
45  interface = "s2-eth2"
46  iperf_file_name = "iperf.json"
47  iperf_flags = ""
48  queue_data_file_name = "qlen.data"
49  plots_dir = "plots_dir_bbr"
```

Листинг 1.3. Конфигурационный файл

Объединение автоматизированных модулей

Имея готовый класс топологии и класс мониторинга, можно создать объекты этих классов, запустить сеть, включить мониторинг сети, построить графики и так далее. Диаграмма активностей для данного приложения представлена на рис. 2. Диаграмма классов разработанного комплекса программ представлена на рис. 3.

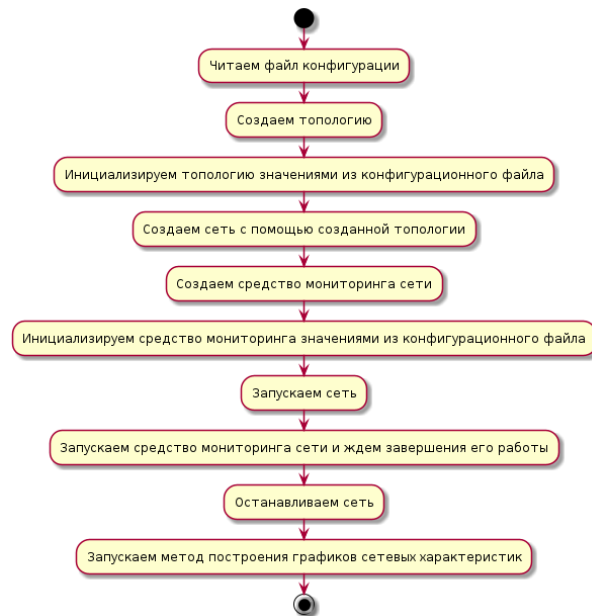


Рис. 2. Диаграмма активностей приложения

Объединение автоматизированных модулей

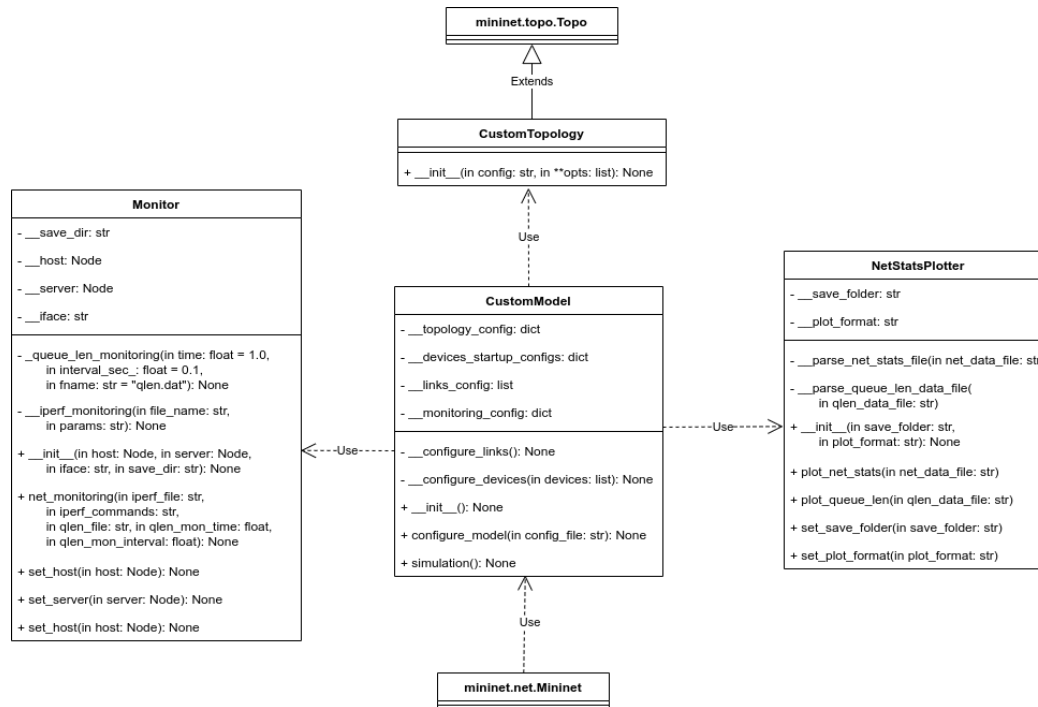


Рис. 3. Диаграмма классов

Объединение автоматизированных модулей

Точка входа в комплекс программ находится в файле `main.py`. Запустить данный скрипт можно с помощью команды

```
sudo ./ main .py -c config / pfifo_config.toml
```

После запуска в каталоге приложения появится директория с именем, которое было указано в `toml`-файле. В ней содержатся графики изменения сетевых характеристик и сырые данные, которые были обработаны объектом класса `NetStatsPlotter`. Например, график изменение длины очереди на интерфейсе `s2-eth2` приведен на рис. 4.

Объединение автоматизированных модулей

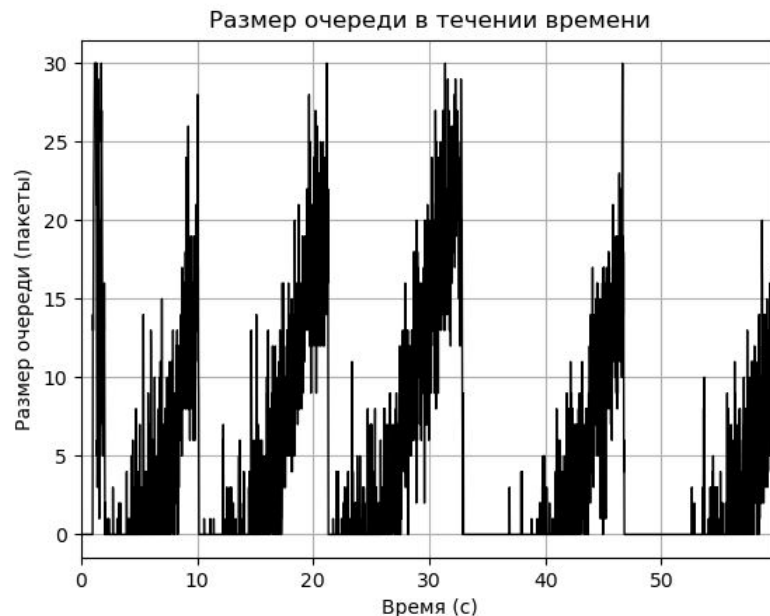


Рис.4. График изменения длины очереди на интерфейсе s2-eth2

Заключение

Анализ работоспособности и производительности сети — неотъемлемая часть работы сетевых инженеров. Современные программы, такие как Mininet, предоставляют разработчикам быстрый и дешевый в построении испытательный полигон, в котором можно проектировать и отлаживать сетевые программы

Литература

1. Mininet. —URL: <http://mininet.org/>.
2. iproute2. —URL: <https://en.wikipedia.org/wiki/Iproute2>.
3. iPerf — the ultimate speed test tool for TCP, UDP and SCTP.—URL: <https://iperf.fr/iperf-doc.php>.
4. Mininet Python API Reference Manual. — URL: <http://mininet.org/api/annotated.html>.
5. TOML.—URL: <https://toml.io/en/>.

