

Информация о процессе и результате работы

Владимир Пуговкин

21 мая 2024 г.

1 Моделирование в NS3

Весь код содержится в файле **NS3-simulation.cpp**. В нем создана конфигурация по документации LTE модуля NS-3. Программа генерирует трафики в UL, DL без затухания, система идеализирована (нет затуханий пакетов, влиний интерференции и прочих радио-эффектов, которые могут возникать в реальных системах связи). Использовался планировщик пакетов PffFMACScheduler. При запуске файлов появляются 4 файла с основными характеристиками, названия которых задаются по умолчанию.

2 Написание скрипта для расчета в UL и DL Troughput

Так как пользователей сети может быть и не только 2, то я решил использовать такую структуру данных как hash-map для быстрого в дальнейшем обращения к данным конкретного пользователя. В данном случае я подумал, что лучше всего использовать данную структуру, потому что обращение происходит за $O(1)$, а также данные сами не будут как-то меняться после их получения. Также hash-map позволит нам раскидать из файлов с RLC характеристиками по каждому пользователю в отдельности, а затем уже работать с этими данными. Например, можно посчитать

Troughput у каждого пользователя. Я считал его следующим образом:

$$Throughput = \frac{ReceivedBytes}{TimeSimulation * 1024} \quad (1)$$

Делил на 1024, чтобы получить скорость в Кб/с. ReceivedBytes - количество байтов, полученных за время симуляции. TimeSimulation - общее время симуляции.

В файле **input-defaults.txt** находятся настройки для симуляции, в которых устанавливается PffF-Mac-scheduler, а также другие параметры по умолчанию, такие как мощность вышки и модель затухания пакетов в пространстве. Также в программе обеспечивается постоянный трафик в UL, DL - пакеты идут постоянно в обе стороны.

Данные я брал из UL,DL traces-файлов, а именно из 4 столбца user-id, 1 и 2 столбца начало и конец i-время симуляции (за какое время дошло i-ое количество байтов), а из 10 столбца полученные байты пользователем.

Функция **main** находится в файле **main.cpp**, файл **Results.cpp** содержит реализацию скрипта, который по полученному выводу ключевых характеристик с RLC уровня считает Throughput каждого пользователя. В файле **Results.hpp** содержатся декларации используемых структур, функций.

3 Флаги запуска компилятора

- Флаги запуска у симуляции в NS-3:

```
1 g++ NS3-simulation.cpp -o NS3-simulation
2 -L/usr/local/lib \
3 -lns3.41-core-default \
4 -lns3.41-lte-default \
5 -lns3.41-network-default \
6 -lns3.41-mobility-default \
7
8 ./NS3-simulation input-defaults.txt
```

- Флаги запуска Скрипта для расчета Throughput каждого пользователя:

```

1  g++ -o main main.cpp Results.cpp
2
3  ./main UlRlcStats.txt DlRlcStats.txt

```

4 Результаты работы скрипта

Скрипт был запущен с двумя пользователями и для примера были взяты следующие настройки mobility для пользователей и вышки:

```

1
2  //coordinates of users (5000 m, 0 m, 0 m), (0 m, 0 m, 0 m)
3
4      UEposition->Add(Vector(5000.0, 0.0, 0.0));
5      UEposition->Add(Vector(0.0, 0.0, 0.0));
6
7  //coordinates of LTE-station (100 m, 100 m, 100 m)
8
9      ENBposition->Add(Vector(100.0, 100.0, 100.0));

```

Результаты работы скрипта при времени симуляции 10 секунд следующие:

```

1  UL stats:
2  USER ID: 1      Throughput: 378.123 Kb/s
3  USER ID: 2      Throughput: 1064.37 Kb/s
4  DL stats:
5  USER ID: 1      Throughput: 663.559 Kb/s
6  USER ID: 2      Throughput: 1084.94 Kb/s

```

```
root@Ubuntu:/home/vlapugb/Desktop/LTE_model_YADRO_task/Task/src# ./main UIRlcSta
ts.txt D1RlcStats.txt
UL stats:
USER ID: 1      Throughput: 378.123 Kb/s
USER ID: 2      Throughput: 1064.37 Kb/s
DL stats:
USER ID: 1      Throughput: 663.559 Kb/s
USER ID: 2      Throughput: 1084.94 Kb/s
```

Рис. 1: Фото из терминала Ubuntu 22.04, вывод throughput в UL, DL для всех пользователей

5 Мини вывод

Для себя я несколько раз запускал свой скрипт с разным количеством пользователей, а также разным расстоянием до вышки. Скрипт выдавал ожидаемые результаты: с ростом числа пользователей росла нагрузка на сеть и тем самым скорость каждого пользователя становилась меньше. Расстояние от вышки до пользователя также играло немаловажную роль, в UL throughput у меня спадал сильнее нежели в DL с расстоянием. Throughput рассчитывался по формуле выше и все результаты были получены по этой формуле.