

Задание для самостоятельного выполнения

Лабораторная работа №4.

Рогожина Н.А.

01 марта 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Рогожина Надежда Александровна
- студентка 3 курса НФИбд-02-22
- Российский университет дружбы народов
- <https://mikogreen.github.io/>

Задание

Описание моделируемой сети: - сеть состоит из N TCP-источников, N TCP-приёмников, двух маршрутизаторов R1 и R2 между источниками и приёмниками (N — не менее 20);

- между TCP-источниками и первым маршрутизатором установлены дуплексные соединения с пропускной способностью 100 Мбит/с и задержкой 20 мс очередью типа DropTail;
- между TCP-приёмниками и вторым маршрутизатором установлены дуплексные соединения с пропускной способностью 100 Мбит/с и задержкой 20 мс очередью типа DropTail;

- между маршрутизаторами установлено симплексное соединение (R1–R2) с пропускной способностью 20 Мбит/с и задержкой 15 мс очередью типа RED, размером буфера 300 пакетов; в обратную сторону — симплексное соединение (R2–R1) с пропускной способностью 15 Мбит/с и задержкой 20 мс очередью типа DropTail;
- данные передаются по протоколу FTP поверх TCP Reno;

- параметры алгоритма RED: $q_{\min} = 75$, $q_{\max} = 150$, $q_w = 0,002$, $p_{\max} = 0.1$;
- максимальный размер TCP-окна 32; размер передаваемого пакета 500 байт; время моделирования — не менее 20 единиц модельного времени.

1. Для приведённой схемы разработать имитационную модель в пакете NS-2.
2. Построить график изменения размера окна TCP (в Xgraph и в GNUPlot).
3. Построить график изменения длины очереди и средней длины очереди на первом маршрутизаторе.
4. Оформить отчёт о выполненной работе.

Теоретическое введение

Network Simulator (NS-2) — один из программных симуляторов моделирования процессов в компьютерных сетях. NS-2 позволяет описать топологию сети, конфигурацию источников и приёмников трафика, параметры соединений (полосу пропускания, задержку, вероятность потерь пакетов и т.д.) и множество других параметров моделируемой системы. Данные о динамике трафика, состоянии соединений и объектов сети, а также информация о работе протоколов фиксируются в генерируемом trace-файле.

Процесс создания модели сети для NS-2 состоит из нескольких этапов:

1. Создание нового объекта класса Simulator, в котором содержатся методы, необходимые для дальнейшего описания модели (например, методы new и delete используются для создания и уничтожения объектов соответственно);
2. Описание топологии моделируемой сети с помощью трёх основных функциональных блоков: узлов (nodes), соединений (links) и агентов (agents);
3. Задание различных действий, характеризующих работу сети.

Выполнение лабораторной работы

Используя данные и код из предыдущих лабораторных работ, создадим новую копию шаблона - `example7.tcl` и оформим задание в виде программы, оставляя комментарии.

Код задания

```
#!/home/openmodelica/tmp/lab-na/example7.ad - Mousepad
# создание объекта Simulator
set ns [new Simulator]

# открытие на запись файла out.nam для визуализатора nam
set nf [open out.nam w]

# все результаты моделирования будут записаны в переменную nf
$ns namtrace-all $nf

# открытие на запись файла трассировки out.tr
# для регистрации всех событий
set f [open out.tr w]

# все регистрируемые события будут записаны в переменную f
$ns trace-all $f

# два узла R1 и R2
set r1 [$ns node]
set r2 [$ns node]

Agent/TCP set window 32
Agent/TCP set packetSize 500

# соединение между узлами и размер очереди
$ns simplex-link $r1 $r2 20Mb 15ms RED
$ns simplex-link $r2 $r1 15Mb 20ms DropTail
$ns queue-limit $r1 $r2 300

# N = 24
set N 24

# добавлен источник и приемник
# FTP потерь TCP Reno
for {set i 0} {$i < $N} {incr i} {
    set n1($i) [$ns node]
    $ns duplex-link $n1($i) $r1 100Mb 20ms DropTail
    set n2($i) [$ns node]
    $ns duplex-link $n2($i) $r2 100Mb 20ms DropTail

    set top($i) [$ns create-connection TCP/Reno $n1($i) TCPSink $n2($i) $i]
    set ftp($i) [$tcp($i) attach-source FTP]
}

# Мониторинг размера окна TCP:
set windowVstTimeFirst [open windowVstTimeRenoFirst w]
puts $windowVstTimeFirst "0.Color: Black"

set windowVstTimeAll [open windowVstTimeRenoAll w]
puts $windowVstTimeAll "0.Color: Black"

set qmon [$ns monitor-queue $r1 $r2 [open qm.out w] 0.1];
$ns link $r1 $r2 queue-sample-timeout;
```

Рис. 1: Код задания

Код задания

```
#!/home/openmodelica/tmp/lab-na/example7.ad - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

# Мониторинг очереди:
set redq [[ss link sr1 sr2] queue]
$redq set thresh 75
$redq set maxthresh 150
$redq set q_weight 0.002
$redq set linterm 10

set tchan [open all.q w]
$redq trace curq
$redq trace auc
$redq attach $tchan_

# Процедура finish:
proc finish () {
    global tchan
    # Подключенный код AWK:
    set awkCode {
        {
            if ($1 == "Q" && NF>2) {
                print $2, $3 >> "temp.q";
                set end $2
            }
            else if ($1 == "a" && NF>2)
                print $2, $3 >> "temp.a";
        }
    }
}

# Если файл есть, то удаляем и создаем новый
exec rm -f temp.q temp.a
exec touch temp.a temp.q

set f [open temp.q w]
puts sf "0.Color: purple"
close sf

set f [open temp.a w]
puts sf "0.Color: blue"
close sf

# выполнение кода AWK
exec awk $awkCode all.q

# Запуск xgraph с графическим окном TCP и очереди:
exec xgraph -bb -bg white -fg black -tk -x time -t "TCP Reno CWND" windowSizeRenoFirst &
exec xgraph -bb -bg white -fg black -tk -x time -t "TCP Reno CWND" windowSizeRenoAll &
exec xgraph -bb -bg white -fg black -tk -x time -y queue temp.q &
exec nan out.nam &
```

Рис. 2: Код задания

```
#!/home/openmodelica/tmp/lab-ns/example7.ad - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

else if ($1 == "q" && NF>2)
    print $2, $3 >> "temp.a";
}

}

# Если файлы есть, то удалим и создам новые
exec rm -f temp.q temp.a
exec touch temp.a temp.q

set f [open temp.q w]
puts $f "0,Color: purple"
close $f

set f [open temp.a w]
puts $f "0,Color: blue"
close $f

# выполнение кода AWK
exec awk $awkCode all.q

# Запуск xgraph с графиком окна TCP и очереди:
exec xgraph -bb -bg white -fg black -tk -x time -t "TCP RenoCWND" windowVsTimeRenoFirst &
exec xgraph -bb -bg white -fg black -tk -x time -t "TCP RenoCWND" windowVsTimeRenoAll &
exec xgraph -bb -bg white -fg black -tk -x time -y queue temp.q &
exec xgraph -bb -bg white -fg black -tk -x time -y queue temp.a &
exec nam out.nam &
exit 0
}

proc plotWindow {tcpSource file} {
    global ns
    set time 0.01
    set now [now now]
    set cwnd [tcpSource set cwnd]
    puts $file "now $cwnd"
    $ns at (expr $now+$time) "plotWindow $tcpSource $file"
}

# ат-событие для планирования событий, которое запускает
# процедуру finish через 5 с после начала моделирования
for (set i 0) ($i < 50) {incr i} {
    $ns at 0.0 "tcp($i) start"
    $ns at 0.0 "plotWindow $tcp($i) $windowVsTimeall"}

$ns at 0.0 "plotWindow $tcp(1) $windowVsTimeFirst"
$ns at 25.0 "finish"

# запуск модели
$ns run
```

Рис. 3: Код задания

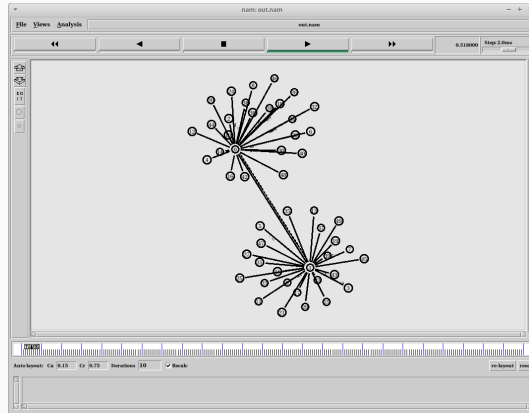


Рис. 4: ns example7.tcl

Код, который нужен для визуализации через GNUPlot мы ввели в самом файле `example7.tcl`.

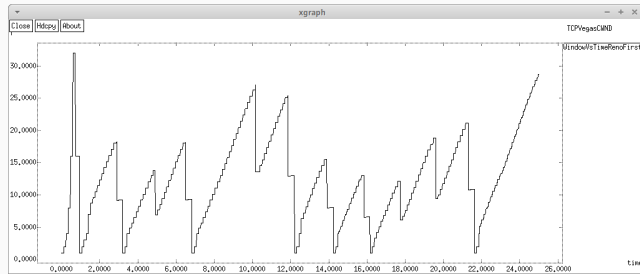


Рис. 5: График изменения размера окна на линке 1-го приемника

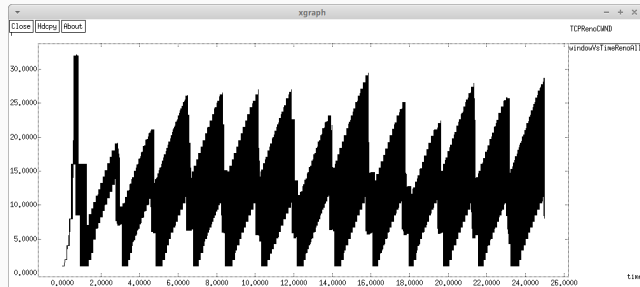


Рис. 6: Изменение размера окна на всех источниках при $N=24$

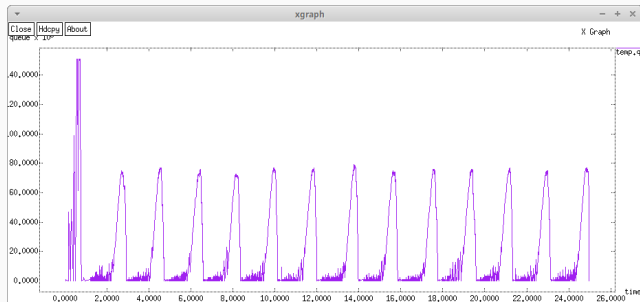


Рис. 7: Фактическая длина очереди

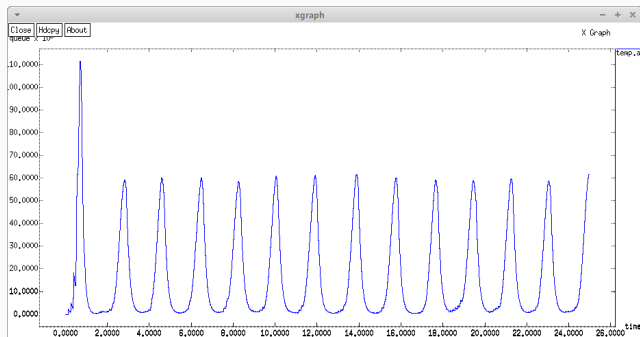
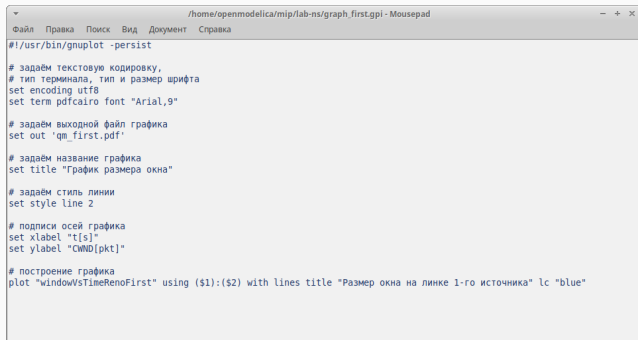


Рис. 8: Средняя длина очереди

Также, нам было необходимо реализовать эти же графики в GNUPlot. Структура всех файлов визуализации одинакова, изменяется только файл, из которого мы берем данные и то, какие колонки участвуют в отрисовке графиков, а также цвет.

A screenshot of a Mousepad window titled "/home/openmodelica/mip/lab-ns/graph_first.gpi - Mousepad". The window contains gnuplot code for visualizing data. The code includes comments in Russian and settings for terminal, output file, title, style, labels, and plot command.

```
#!/usr/bin/gnuplot -persist

# задаём текстовую кодировку,
# тип терминала, тип и размер шрифта
set encoding utf8
set term pdfcairo font "Arial,9"

# задаём выходной файл графика
set out 'qm_first.pdf'

# задаём название графика
set title "График размера окна"

# задаём стиль линии
set style line 2

# подписи осей графика
set xlabel "t[s]"
set ylabel "CWND[pkt]"

# построение графика
plot "windowVsTimeRenoFirst" using ($1):($2) with lines title "Размер окна на линке 1-го источника" lc "blue"
```

Рис. 9: Код визуализации xgraph

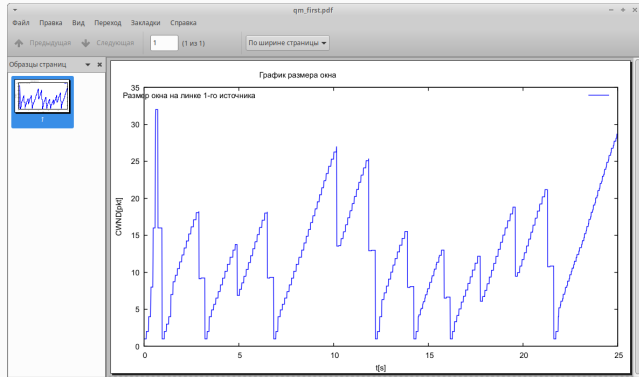


Рис. 10: График изменения размера окна на линке 1-го приемника

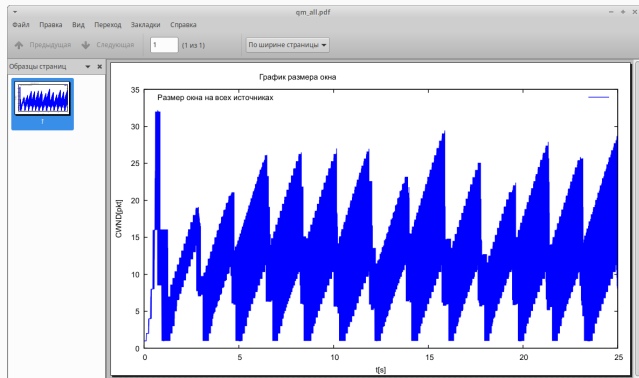


Рис. 11: Изменение размера окна на всех источниках при $N=24$

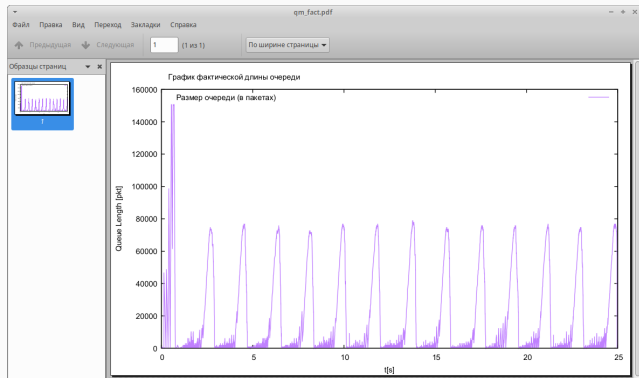


Рис. 12: Фактическая длина очереди

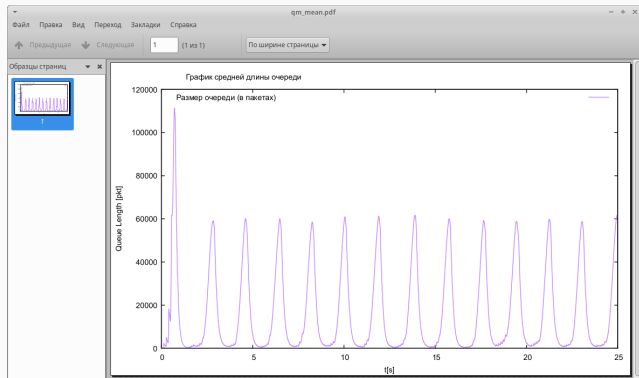


Рис. 13: Средняя длина очереди

Выводы

В ходе лабораторной работы мы реализовали схему, указанную в задании, а также визуализировали основные характеристики очереди с помощью 2-х инструментов визуализации.