

# **Отчёт по лабораторной работе №4**

**Задание для самостоятельного выполнения**

Надежда Александровна Рогожина

# Содержание

<b>1</b>	<b>Задание</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

3.1	Код задания . . . . .	7
3.2	Код задания . . . . .	8
3.3	Код задания . . . . .	9
3.4	ns example7.tcl . . . . .	10
3.5	График изменения размера окна на линке 1-го приемника . . . . .	10
3.6	Изменение размера окна на всех источниках при N=24 . . . . .	11
3.7	Фактическая длина очереди . . . . .	11
3.8	Средняя длина очереди . . . . .	11
3.9	Код визуализации xgraph . . . . .	12
3.10	График изменения размера окна на линке 1-го приемника . . . . .	12
3.11	Изменение размера окна на всех источниках при N=24 . . . . .	13
3.12	Фактическая длина очереди . . . . .	13
3.13	Средняя длина очереди . . . . .	14

## **Список таблиц**

# 1 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.

**Задание:** 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.

## 2 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.

Более подробно про NS-2 см. в [1].

## 3 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 (рис. 3.1, рис. 3.2, рис. 3.3).

Рис. 3.1: Код задания

```

# Мониторинг очереди:
set redq [[${ns link $r1 $r2} 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 ave
$redq attach $tchan_

# Процедура finish:
proc finish {} {
    global tchan_
    # подключение кода 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 файлы есть, то удаляем и создаем 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 графиками окна TCP и очереди:
exec xgraph -bb -bg white -fg black -tk -x time -t "TCPReNoCWND" windowVsTimeReNoFirst &
exec xgraph -bb -bg white -fg black -tk -x time -t "TCPReNoCWND" 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 &

```

Рис. 3.2: Код задания



```

/home/openmodelica/mip/lab-ns/example7.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

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

# Если файлы есть, то удаляем и создаем 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 графиками окна TCP и очереди:
exec xgraph -bb -bg white -fg black -tk -x time -t "TCPReNoCWND" windowVsTimeRenoFirst &
exec xgraph -bb -bg white -fg black -tk -x time -t "TCPReNoCWND" 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 [$ns now]
    set cwnd [$tcpSource set cwnd_]
    puts $file "$now $cwnd"
    $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

# at-событие для планировщика событий, которое запускает
# процедуру finish через 5 с после начала моделирования
for {set i 0} {$i < $N} {incr i} {
    $ns at 0.0 "$ftp($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. 3.3: Код задания

Запустим моделирование процесса (рис. 3.4).

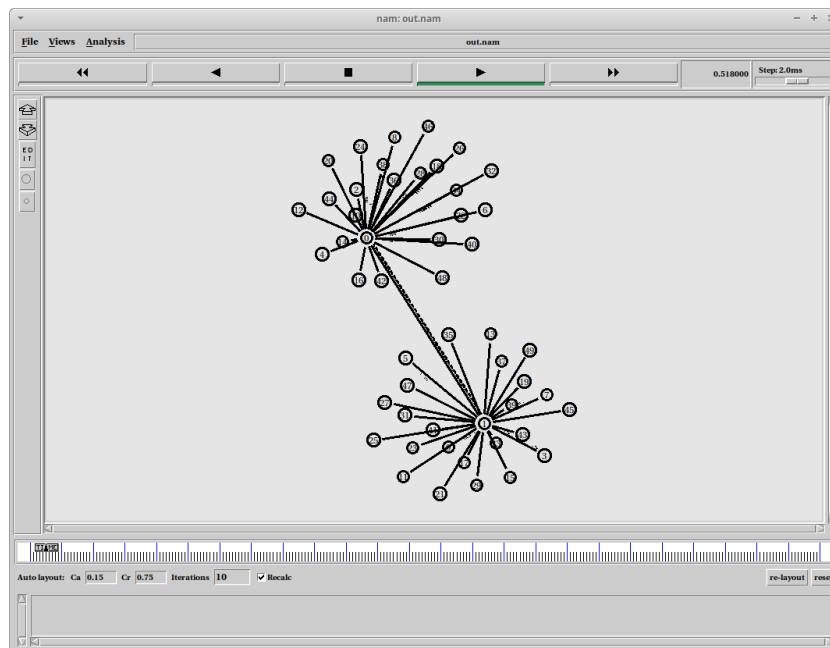


Рис. 3.4: ns example7.tcl

Код, который нужен для визуализации через GNUPlot мы ввели в самом файле example7.tcl. Результаты (рис. 3.5, рис. 3.6, рис. 3.7, рис. 3.8):

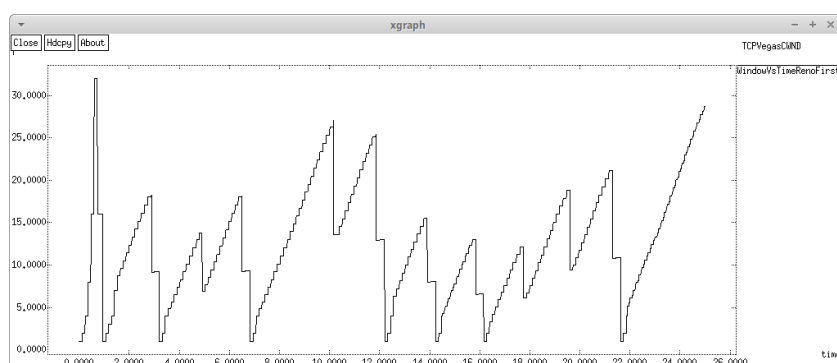


Рис. 3.5: График изменения размера окна на линке 1-го приемника

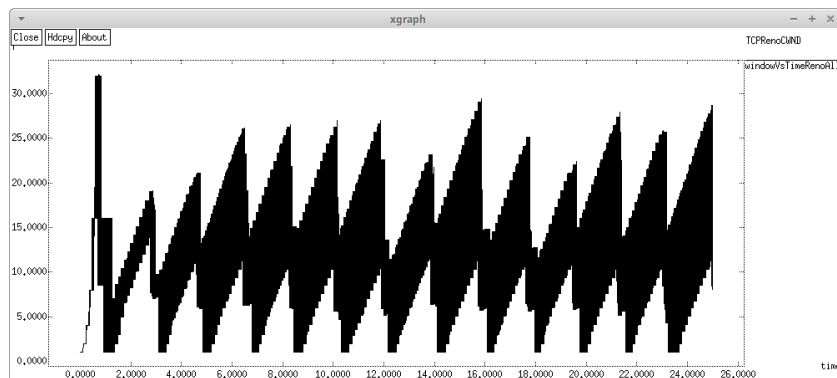


Рис. 3.6: Изменение размера окна на всех источниках при  $N=24$

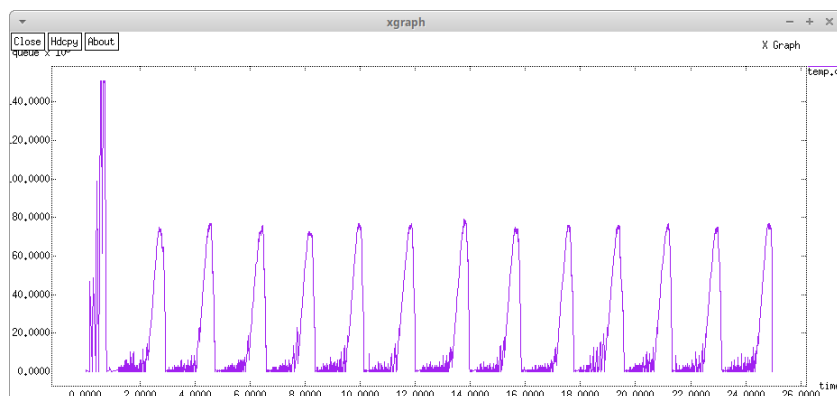


Рис. 3.7: Фактическая длина очереди

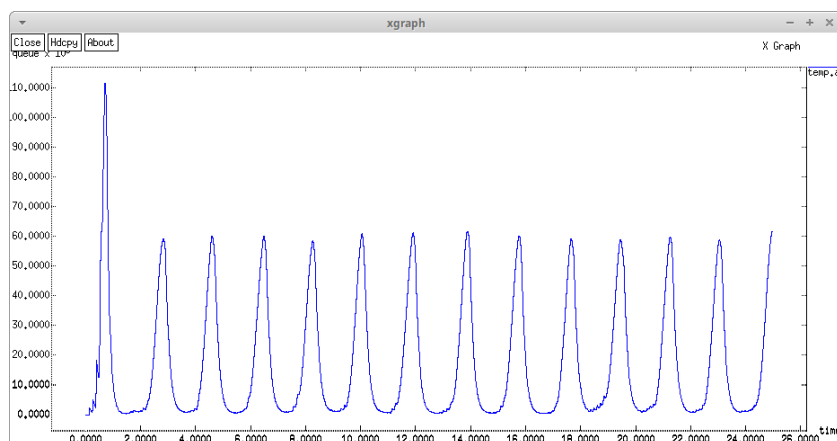


Рис. 3.8: Средняя длина очереди

Также, нам было необходимо реализовать эти же графики в xgraph. Структура всех файлов визуализации одинакова, изменяется только файл, из которого мы

берем данные и то, какие колонки участвуют в отрисовке графиков, а также цвет (рис. 3.9).

```

/home/openmodelica/mip/lab-ns/graph_first.gpi - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
#!/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. 3.9: Код визуализации xgraph

Таким образом, мы получили отображение тех же данных другим способом визуализации (рис. 3.10, рис. 3.11, рис. 3.12, рис. 3.13).

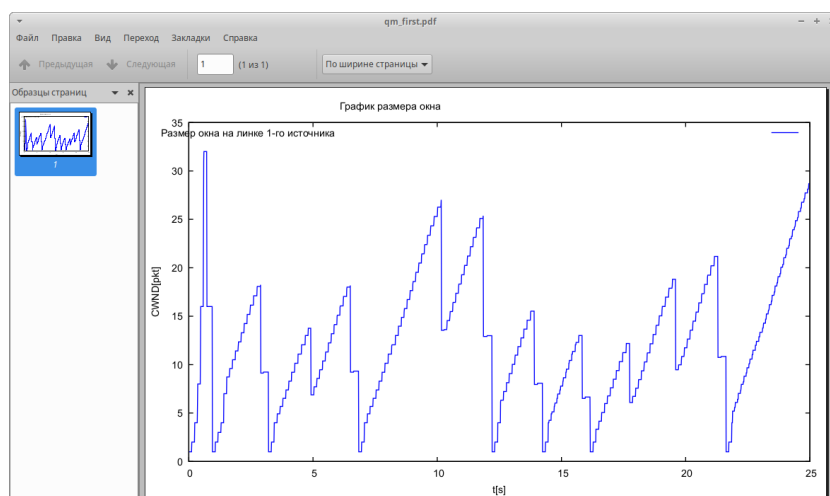


Рис. 3.10: График изменения размера окна на линке 1-го приемника

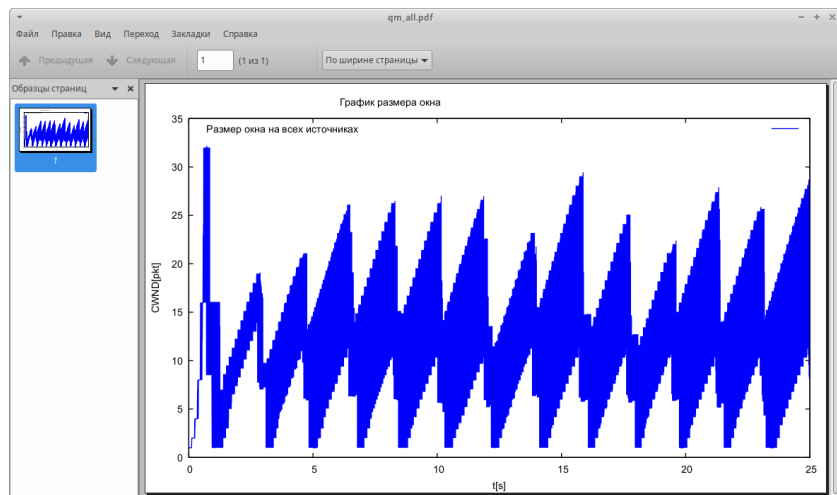


Рис. 3.11: Изменение размера окна на всех источниках при  $N=24$

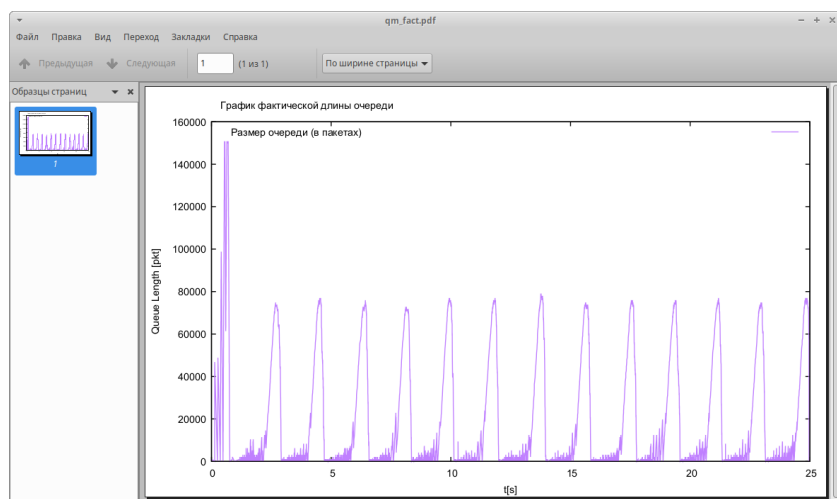


Рис. 3.12: Фактическая длина очереди

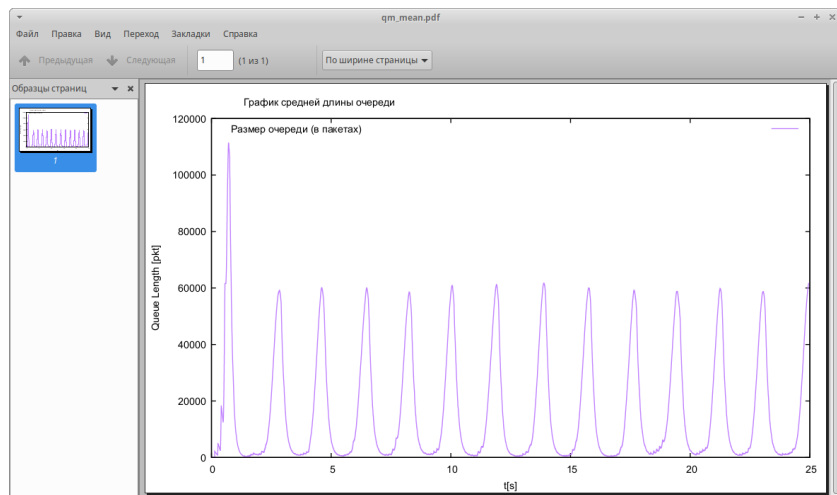


Рис. 3.13: Средняя длина очереди

## 4 Выводы

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

## **Список литературы**

1. А. В. Королькова Д.С.К. Моделирование информационных процессов. 1-е изд. Москва: Типография РУДН, 2014. 191 с.