

Лабораторная работа №4

Имитационное моделирование

Екатерина Канева, НФИбд-02-22

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Теоретическое введение | 7 |
| 4 | Выполнение лабораторной работы | 8 |
| 5 | Выводы | 21 |

Список иллюстраций

| | | |
|-----|--|----|
| 4.1 | Схема в NS-2. | 12 |
| 4.2 | Изменение размера окна TCP на линке 1 источника при $N = 25$. . . | 13 |
| 4.3 | Изменение размера окна TCP на всех источниках при $N = 25$. . . | 14 |
| 4.4 | Изменение длины очереди на линке (R1-R2). | 15 |
| 4.5 | Изменение средней длины очереди на линке (R1-R2). | 16 |
| 4.6 | Изменение размера окна TCP на линке 1 источника при $N = 25$. . . | 18 |
| 4.7 | Изменение размера окна TCP на всех источниках при $N = 25$. . . | 19 |
| 4.8 | Изменение длины очереди на линке (R1-R2). | 19 |
| 4.9 | Изменение средней длины очереди на линке (R1-R2). | 20 |

Список таблиц

1 Цель работы

Выполнить задание для самостоятельной работы.

2 Задание

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

3 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.

4 Выполнение лабораторной работы

Нужно было смоделировать следующую сеть:

- сеть состоит из N ТСП-источников, N ТСП-приёмников, двух маршрутизаторов R1 и R2 между источниками и приёмниками (N — не менее 20);
- между ТСП-источниками и первым маршрутизатором установлены дуплексные соединения с пропускной способностью 100 Мбит/с и задержкой 20 мс очередью типа DropTail;
- между ТСП-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 ТСП-окна 32; размер передаваемого пакета 500 байт; время моделирования — не менее 20 единиц модельного времени.

Я реализовала следующий код программы, выбрав $N = 25$ и время работы 25.0:

```
set ns [new Simulator]
set nf [open out.nam w]
```



```
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f
```

```
Agent/TCP set window_ 32
Agent/TCP set pktSize_ 500
```

```
proc finish {} {
    global tchan_
    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";
    }}
}
```

```
exec rm -f temp.q temp.a
exec touch temp.a temp.q
```

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

```
set f [open temp.a w]
puts $f "1.Color: White"
close $f
```

```

exec awk $awkCode all.q

exec xgraph -fg blue -bg white -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno0
exec xgraph -fg blue -bg white -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno0
exec xgraph -bb -tk -x time -y queue temp.q &
exec xgraph -bb -tk -x time -y queue temp.a &
exec nam out.nam &
exit 0
}

# Формирование файла с данными о размере окна TCP:
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"
}

# Узлы сети:
set r1 [$ns node]
set r2 [$ns node]

$ns simplex-link $r1 $r2 20Mb 15ms RED
$ns simplex-link $r2 $r1 15Mb 20ms DropTail
$ns queue-limit $r1 $r2 300

set N 25

```

```

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) $r2 100Mb 20ms DropTail
    set tcp($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 windowVsTimeOne [open WindowVsTimeRenoOne w]
puts $windowVsTimeOne "0.Color: Blue"
set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "1.Color: Blue"

```

```

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

```

```

# 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_

```

```

for {set i 0} {$i < $N} {incr i} {
    $ns at 0.0 "$ftp($i) start"
    $ns at 0.0 "plotWindow $tcp($i) $windowVsTime"
}

```

Добавление at-событий:

```

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

```

После чего запустила программу и получила следующую схему в NS-2, она работала (рис. 4.1):

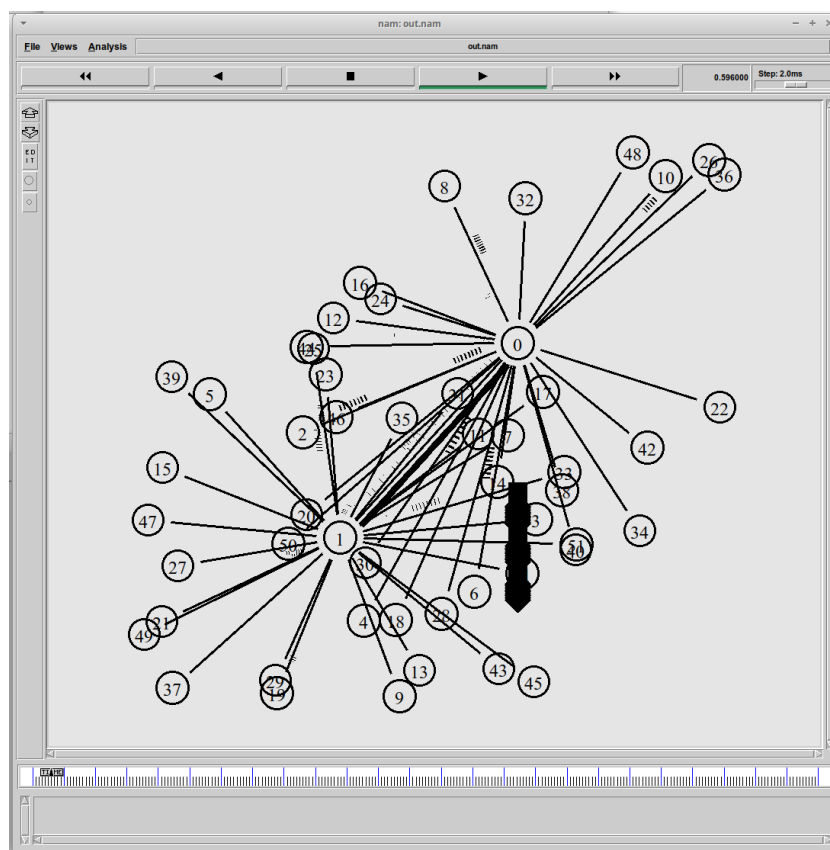


Рис. 4.1: Схема в NS-2.

Также вывелись 4 графика (рис. 4.2, 4.3, 4.4 и 4.5):

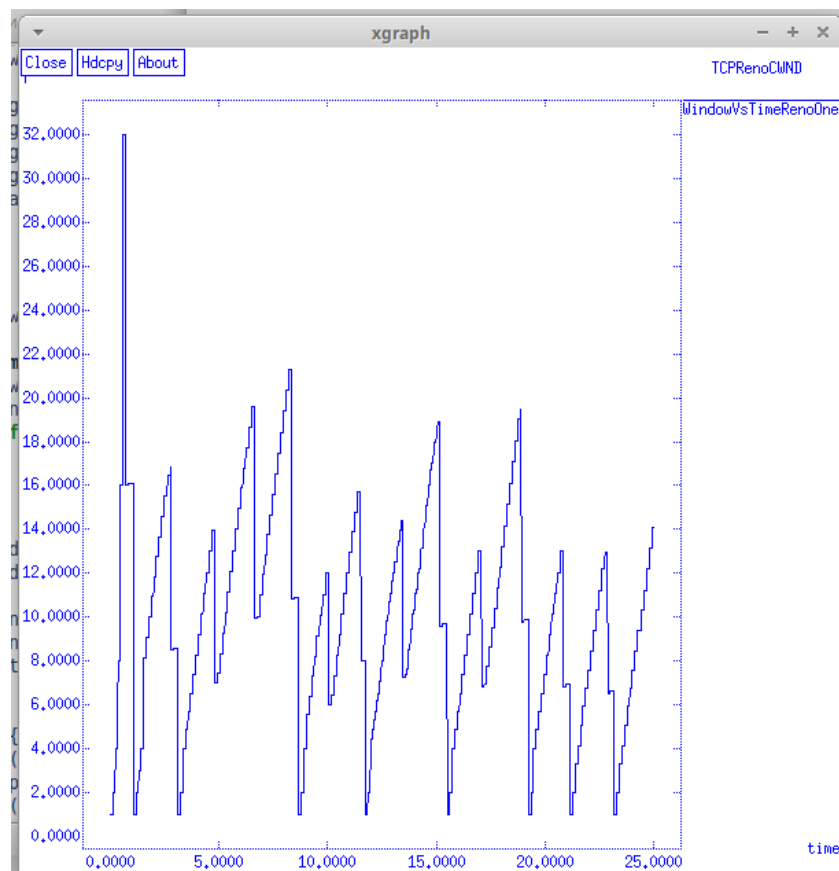


Рис. 4.2: Изменение размера окна TCP на линке 1 источника при $N = 25$.

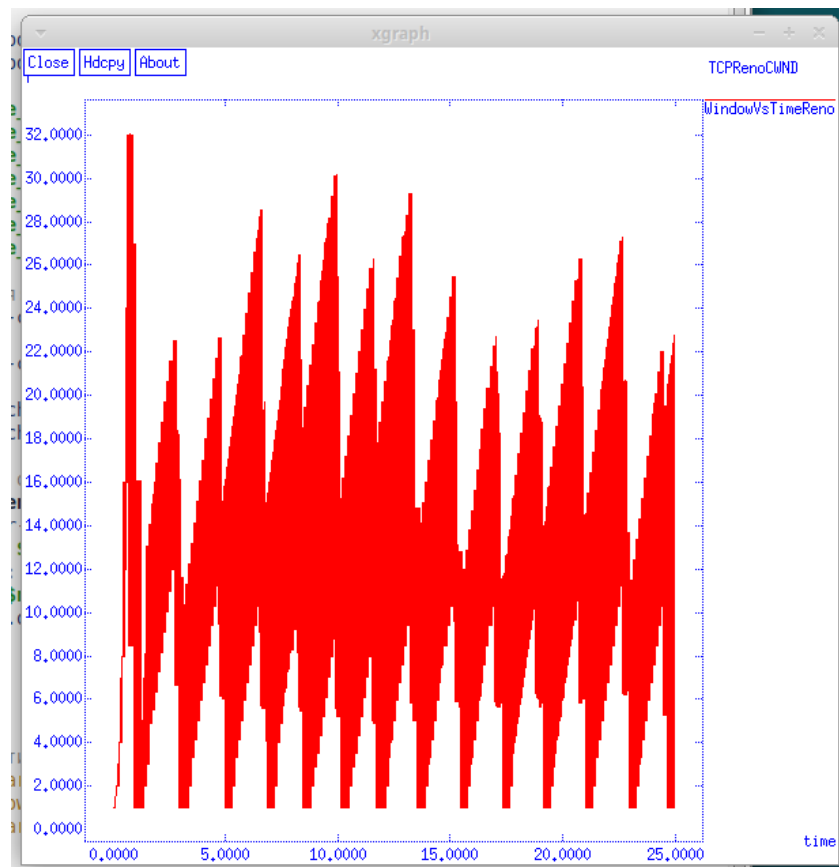


Рис. 4.3: Изменение размера окна TCP на всех источниках при $N = 25$.

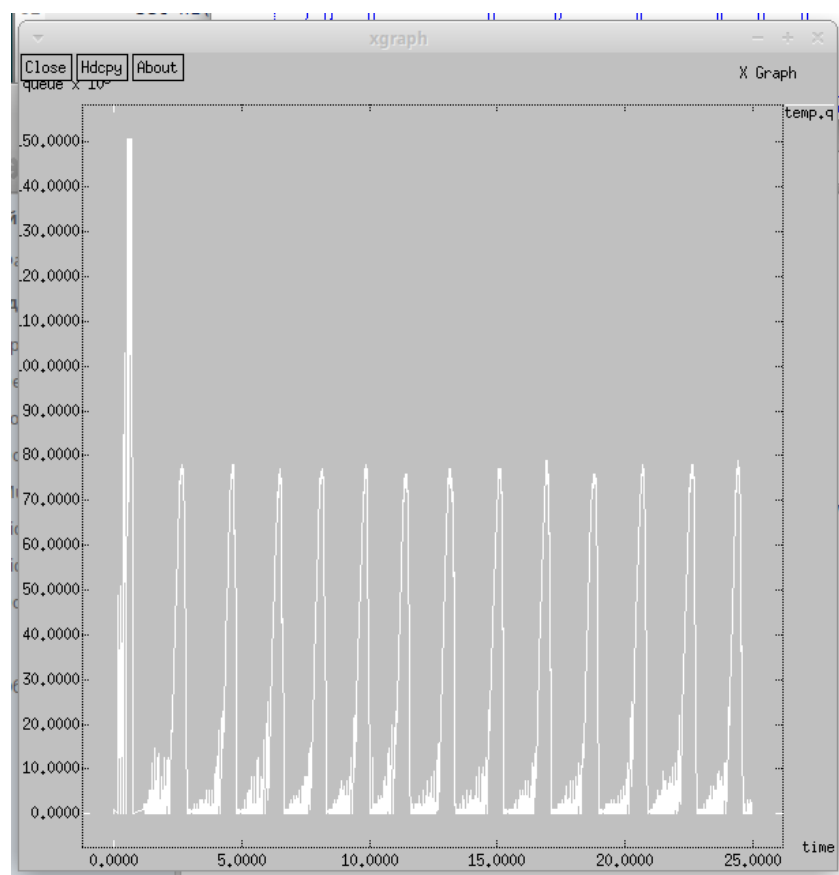


Рис. 4.4: Изменение длины очереди на линке (R1-R2).

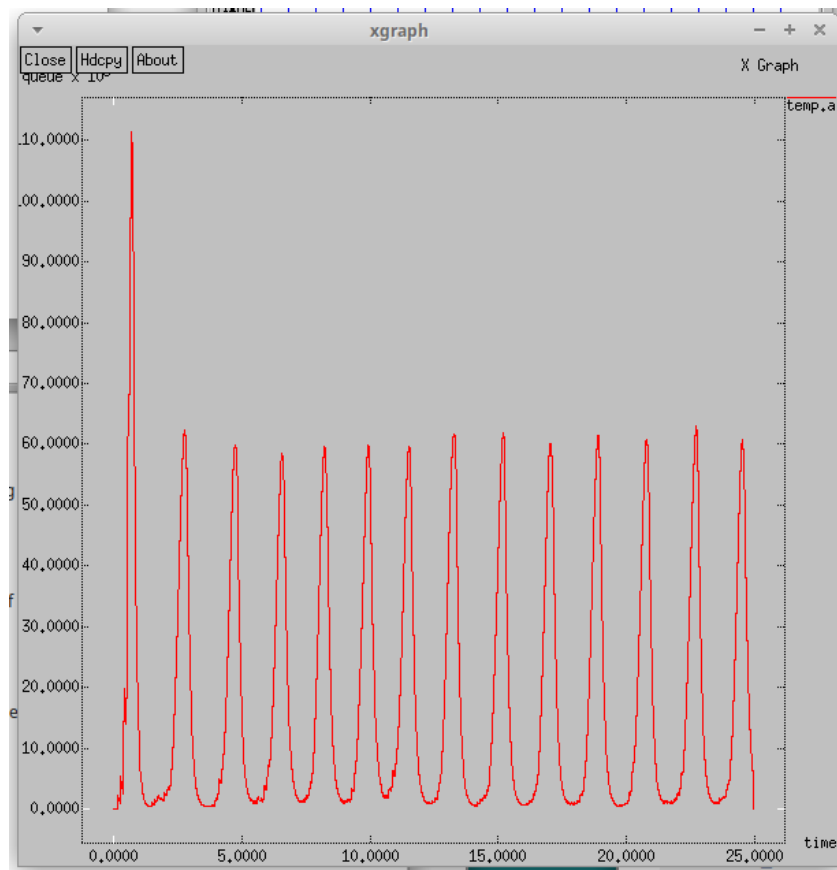


Рис. 4.5: Изменение средней длины очереди на линке (R1-R2).

Далее я реализовала код, строящий графики в GNUPlot:

```
#!/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 '1_tcp.pdf'

# задаём название графика
```



```

set title "Изменение размера окна TCP на линке 1 источника при N = 25"

# подписи осей
set xlabel "t[s]" font "Arial,9"
set ylabel "CWND [pkt]" font "Arial,9"

# построение графика, используя значения
# 1-го и 2-го столбцов файла вывода
plot "WindowVsTimeRenoOne" using ($1):($2) with lines title "Размер окна TCP"

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

# задаём название графика
set title "Изменение размера окна TCP на всех источниках при N = 25"

# построение графика, используя значения
# 1-го и 2-го столбцов файла вывода
plot "WindowVsTimeReno" using ($1):($2) with lines title "Размер окна TCP"

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

# задаём название графика
set title "Изменение длины очереди на линке (R1-R2)"

# подпись к оси
set ylabel "Queue length [pkt]" font "Arial,9"

```

```

# построение графика, используя значения
# 1-го и 2-го столбцов файла вывода
plot "temp.q" using ($1):($2) with lines title "Длина очереди"

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

# задаём название графика
set title "Изменение средней длины очереди на линке (R1-R2)"

# подпись к оси
set ylabel "Average queue length [pkt]" font "Arial,9"

# построение графика, используя значения
# 1-го и 2-го столбцов файла вывода
plot "temp.a" using ($1):($2) with lines title "Средняя длина очереди"

```

Сделала файл исполняемым с помощью команды `chmod +x graph_plot`, при запуске создались следующие графики в файлах PDF (рис. 4.6, 4.7, 4.8 и 4.9):

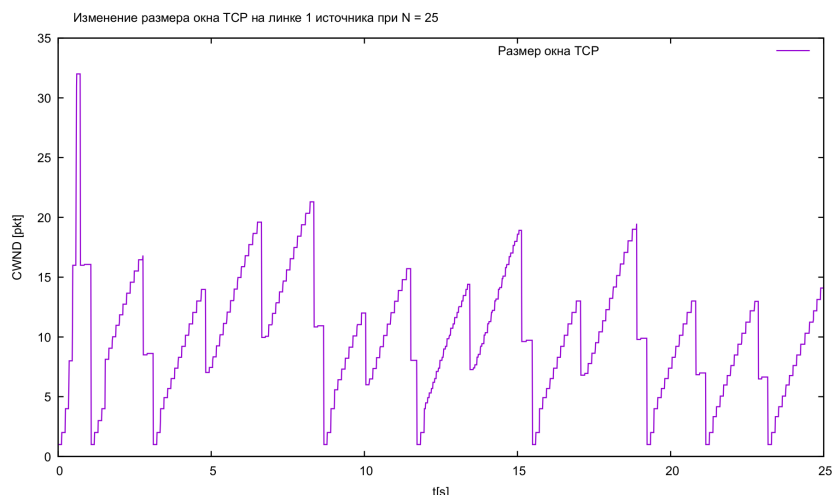


Рис. 4.6: Изменение размера окна TCP на линке 1 источника при N = 25.

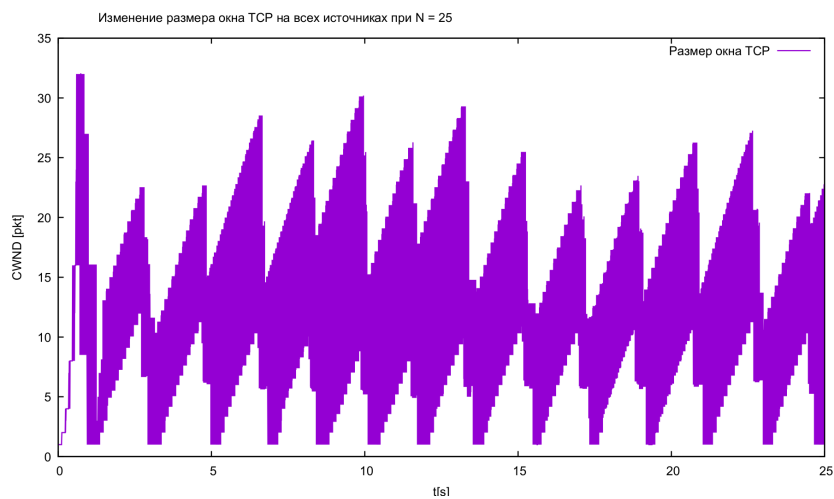


Рис. 4.7: Изменение размера окна TCP на всех источниках при N = 25.

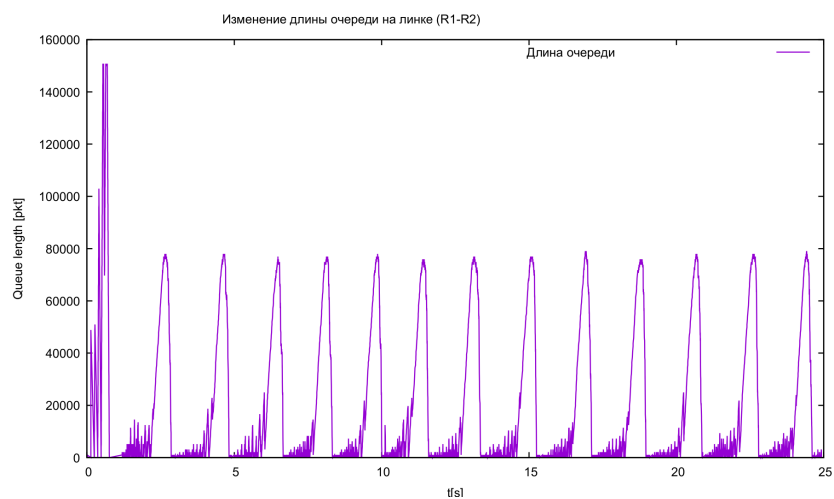


Рис. 4.8: Изменение длины очереди на линке (R1-R2).

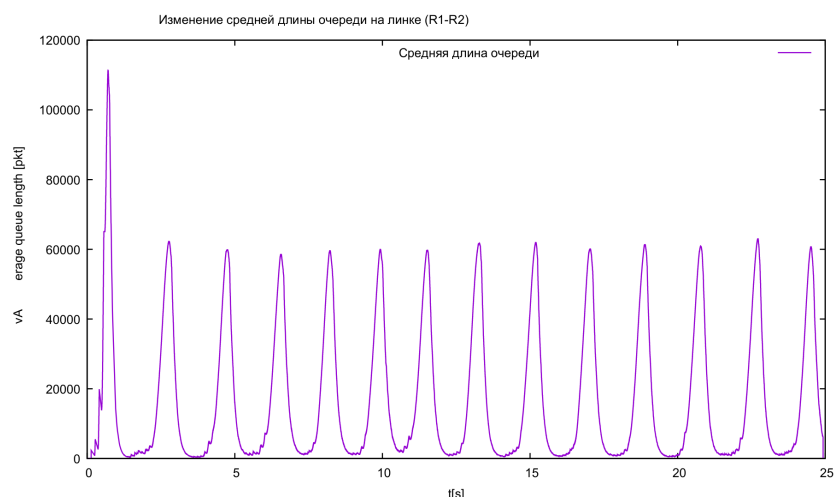


Рис. 4.9: Изменение средней длины очереди на линке (R1-R2).

5 Выводы

Выполнила задание для самостоятельной работы.