

Лабораторная работа №4.

Задание для самостоятельного выполнения

Александр Андреевич Шуплецов

Содержание

1	Цель работы	5
2	Выполнение работы	6
3	Выводы	17
	Список литературы	18

Список иллюстраций

2.1	график 1	10
2.2	график 2	11
2.3	график 3	12
2.4	график 4	13
2.5	график в GnUpot 1	15
2.6	график в GnUpot 2	15
2.7	график в GnUpot 3	16
2.8	график в GnUpot 4	16

Список таблиц

1 Цель работы

Приобретение навыков моделирования стохастических процессов с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования, сделать практическое задание.

2 Выполнение работы

1. Выполним задание для самостоятельного выполнения в файле lab4.tcl.

```
#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
```

```
# 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";
    }
}

```

```

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

# выполнение кода AWK
exec awk $awkCode all.q

```

```

# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeRenoAll &
exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno1 &
exec xgraph -bb -tk -x time -y queue temp.q &
exec xgraph -bb -tk -x time -y ave_queue temp.a &
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 $file"
}
# маршрутизаторы
set node_(r1) [$ns node]
set node_(r2) [$ns node]
# Соединения:
$ns simplex-link $node_(r1) $node_(r2) 20Mb 15ms RED
$ns simplex-link $node_(r2) $node_(r1) 15Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 300
set N 25
for {set i 0} {$i < $N} {incr i} {
    set node_(s$i) [$ns node]
    $ns duplex-link $node_(s$i) $node_(r1) 100Mb 20ms DropTail
    set node_(d$i) [$ns node]
    $ns duplex-link $node_(d$i) $node_(r2) 100Mb 20ms DropTail

    set tcp_($i) [$ns create-connection TCP/Reno $node_(s$i) TCPSink $node_(d$i)]
    $tcp_($i) set window_ 32
    $tcp_($i) set pktSize_ 500
    set ftp_($i) [$tcp_($i) attach-source FTP]
}

# Мониторинг размера окна TCP:
set windowVsTimeAll [open WindowVsTimeRenoAll w]
set windowVsTime1 [open WindowVsTimeReno1 w]

set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[$ns link $node_(r1) $node_(r2)] queue-sample-timeout;
# Мониторинг очереди:

```



```

set redq [[ $ns link $node_(r1) $node_(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_

#at-событие для планировщика событий, которое запускает
#процедуру finish через 5 с после начала моделирования
# Добавление at-событий:
for {set i 0} {$i < $N} {incr i} {
    $ns at 0.0 "$ftp_($i) start"
    $ns at 1.1 "plotWindow $tcp_($i) $windowVsTimeAll"
}
$ns at 1.1 "plotWindow $tcp_(1) $windowVsTime1"

$ns at 25 "finish"
#запуск модели
$ns run

```

2. Проанализируем результаты полученных графиков.

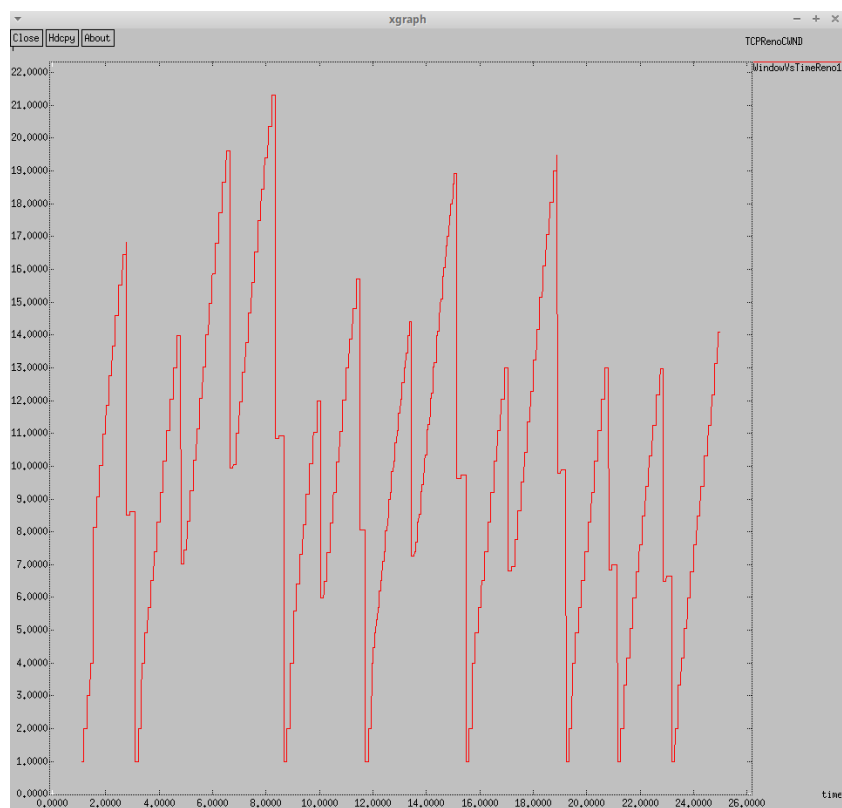


Рис. 2.1: график 1

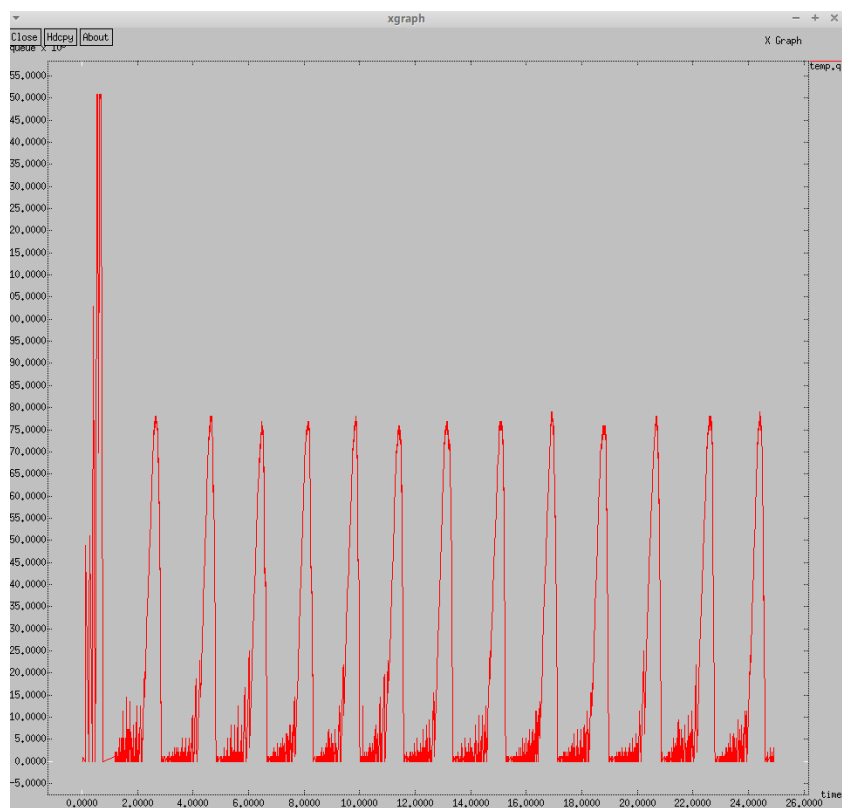


Рис. 2.2: график 2

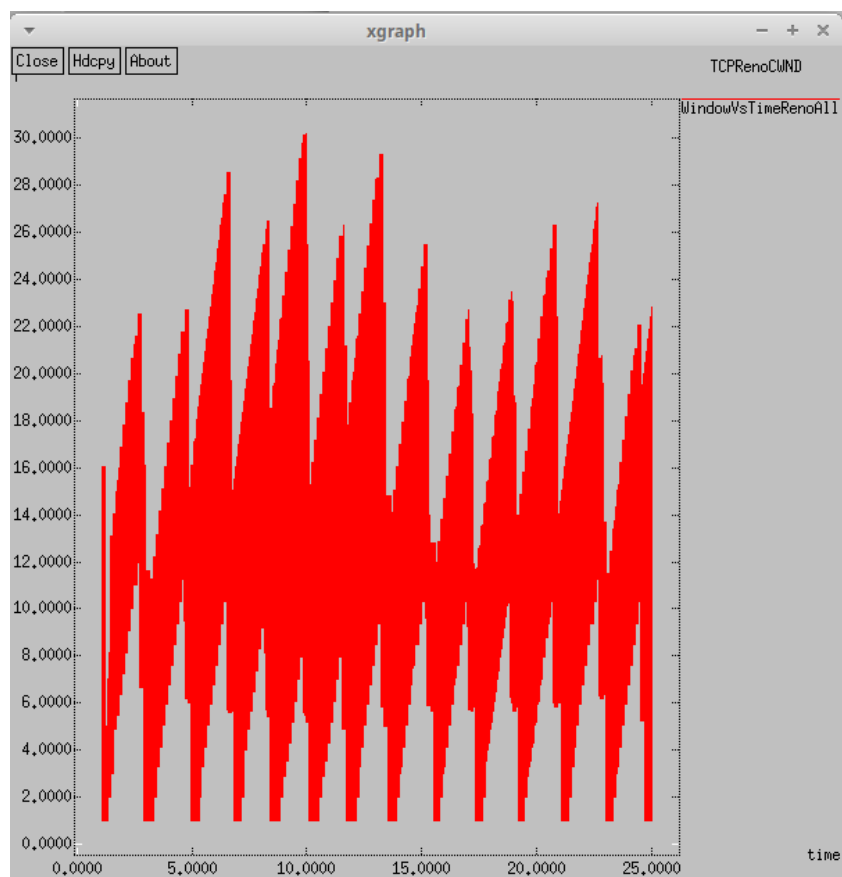


Рис. 2.3: график 3

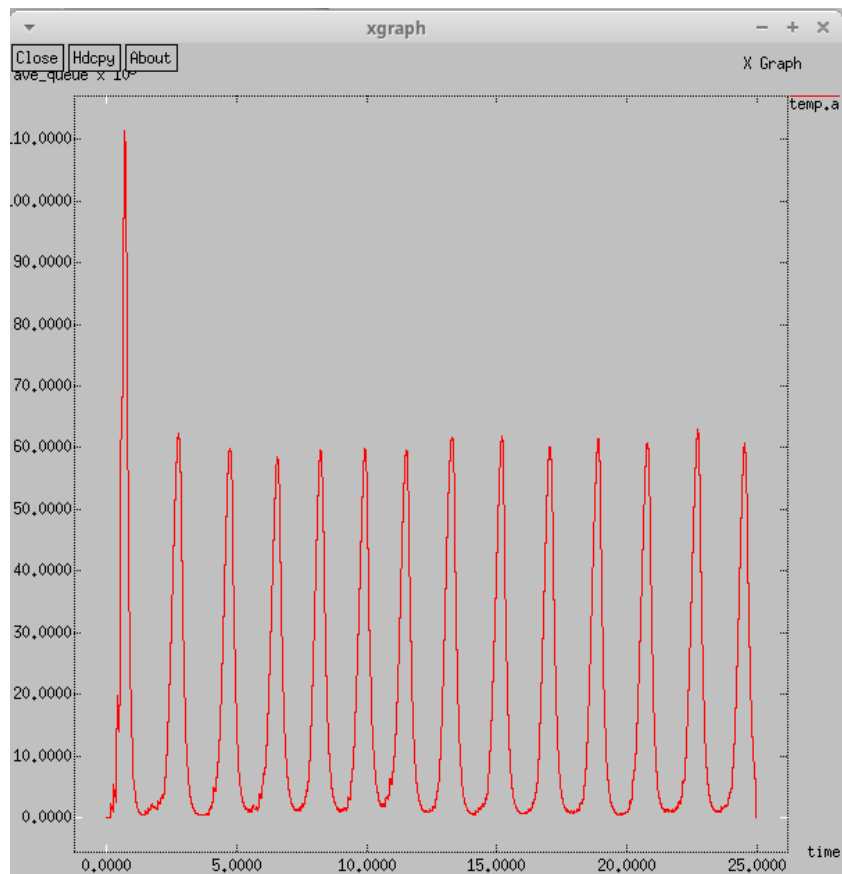


Рис. 2.4: график 4

3. Построим графики в GnUpplot с помощью данного кода.

```
#!/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 'WvsT1.pdf'
# задаём название графика
set title "Изменение размера окна TCP на линке 1-го источника при N=20"
# задаём стиль линии
```

```

set style line 2
# подписи осей графика
set xlabel "t[s]"
set ylabel "CWND [pkt]"
# построение графика, используя значения
plot "WindowVsTimeReno1" using ($1):($2) with lines title "Размер окна TCP"

# задаём выходной файл графика
set out 'WvsTAll.pdf'
# задаём название графика
set title "Изменение размера окна TCP на всех источниках при N=20"
# построение графика, используя значения
plot "WindowVsTimeRenoAll" using ($1):($2) with lines title "Размер окна TCP"

# задаём выходной файл графика
set out 'queue.pdf'
# задаём название графика
set title "Изменение размера длины очереди на линке (R1-R2) при N=20, qmin = 75,"
# подписи осей графика
set xlabel "t[s]"
set ylabel "Queue Length [pkt]"
# построение графика, используя значения
plot "temp.q" using ($1):($2) with lines title "Длина очереди"

# задаём выходной файл графика
set out 'ave_queue.pdf'
# задаём название графика
set title "Изменение размера средней длины очереди на линке (R1-R2) при N=20, qmi"
# подписи осей графика

```

```

set xlabel "t[s]"
set ylabel "Queue Length [pkt]"
# построение графика, используя значения
plot "temp.a" using ($1):($2) with lines title "Средняя длина очереди"

```

4. Проанализируем результаты графиков в GNUplot.

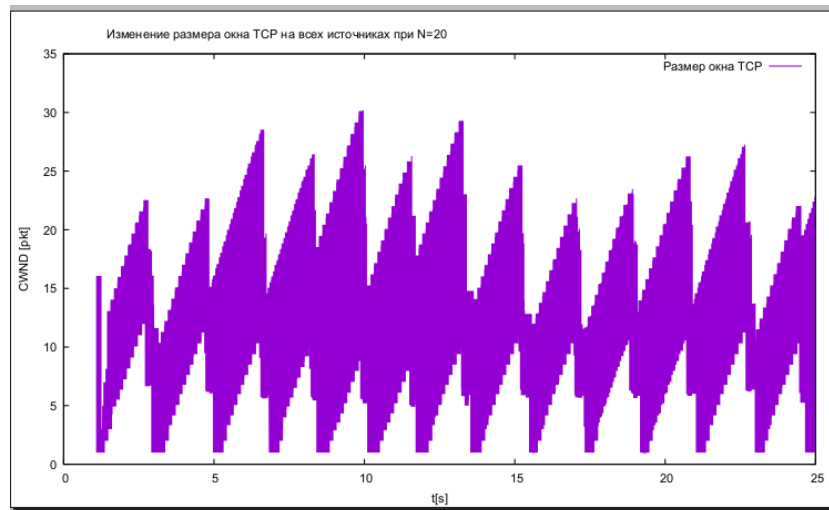


Рис. 2.5: график в GNUplot 1

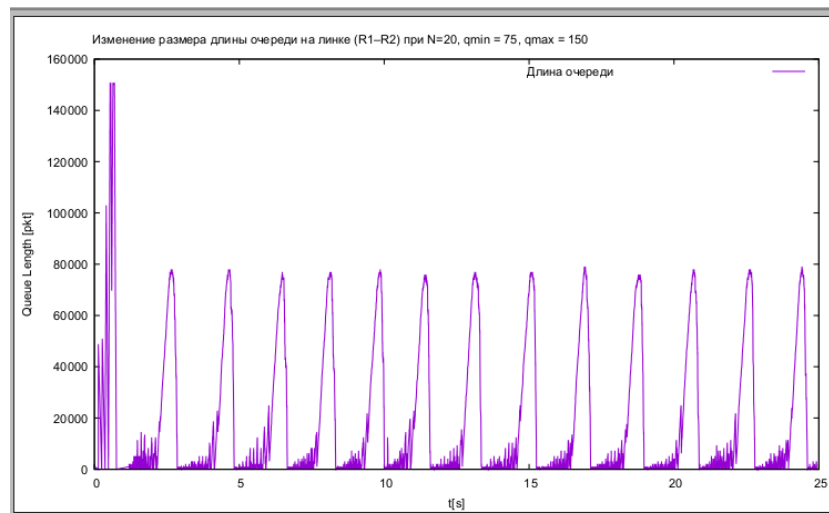


Рис. 2.6: график в GNUplot 2

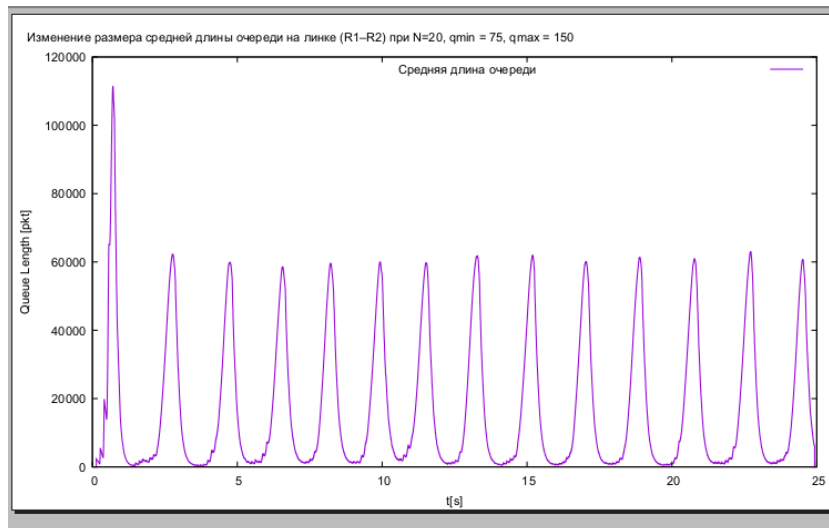


Рис. 2.7: график в GNUplot 3

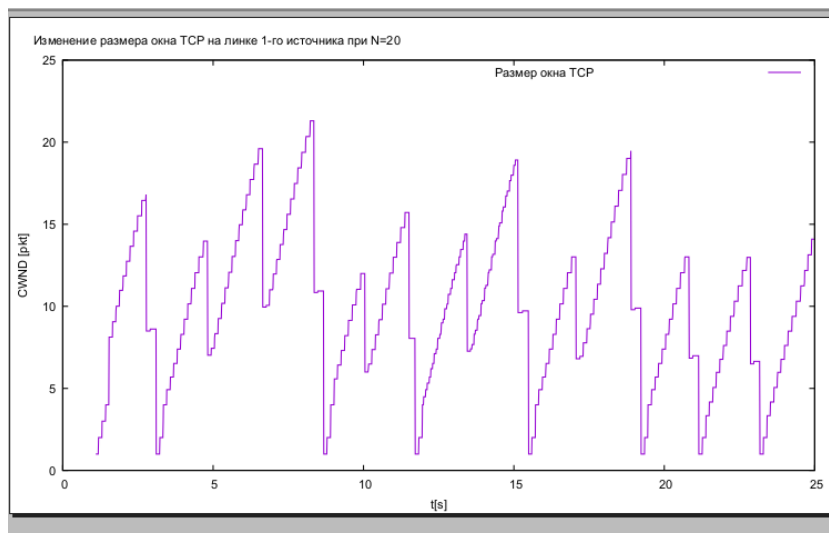


Рис. 2.8: график в GNUplot 4

3 Выводы

Я приобрел навыки моделирования стохастических процессов с помощью средства имитационного моделирования NS-2, а также сделал анализ полученных результатов моделирования, сделал практическое задание.

Список литературы

Королькова А. В., Кулябов Д.С. “Материалы к лабораторным работам”