

# **Лабораторная работа 4**

**Имитационное моделирование**

Оразгелдиев Язгелди

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>21</b>
	<b>Список литературы</b>	<b>22</b>

# Список иллюстраций

3.1	Схема моделируемой сети при $N=30$ . . . . .	12
3.2	Изменение размера окна ТСР на всех источниках (xgraph) . . . . .	13
3.3	Изменение размера окна ТСР на линке 1-го источника(xgraph) . . . . .	14
3.4	Изменение размера длины очереди на линке $(R1-R2)$ (xgraph) . . . . .	15
3.5	Изменение размера средней длины очереди на линке $(R1-R2)$ (xgraph) . . . . .	16
3.6	Изменение размера окна ТСР на всех источниках (GNU) . . . . .	19
3.7	Изменение размера окна ТСР на линке 1-го источника(GNU) . . . . .	19
3.8	Изменение размера длины очереди на линке $(R1-R2)$ (GNU) . . . . .	20
3.9	Изменение размера средней длины очереди на линке $(R1-R2)$ (GNU) . . . . .	20

## **Список таблиц**

# **1 Цель работы**

Выполнить задание для самостоятельного выполнения и закрепить материал, изученный ранее.

## 2 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.

### 3 Выполнение лабораторной работы

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:  $q_{min} = 75$ ,  $q_{max} = 150$ ,  $qw = 0,002$ ,  $r_{max} = 0.1$ ; – максимальный размер TCP-окна 32; размер передаваемого пакета 500 байт; время моделирования – не менее 20 единиц модельного времени

Создали файл lab4.tcl, написали в нём код для реализации приведенной моделируемой сети при  $N=30$  и временем моделирования 30 единиц

```
# 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
```

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

```
# 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
```

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

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

```
exec awk $awkCode all.q
```

```
# Запуск xgraph с графиками окна TCP и очереди:
```

```
exec xgraph -fg pink -bg purple -bb -tk -x time -t "TCPReoCWND" WindowVsTimeRenoOne &
exec xgraph -fg pink -bg purple -bb -tk -x time -t "TCPReoCWND" WindowVsTimeRenoAll &
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"
```

```

}

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 30
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: White"
set windowVsTimeAll [open WindowVsTimeRenoAll w]
puts $windowVsTimeAll "0.Color: White"

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) $windowVsTimeAll"
}

$ns at 0.0 "plotWindow $tcp(1) $windowVsTimeOne"

# at-событие для планировщика событий, которое запускает
# процедуру finish через 20s после начала моделирования
$ns at 20.0 "finish"

# запуск модели
$ns run

```

После запуска нашей программы получили схему модели в визуализаторе `nam`. Если запустить ее, то можно увидеть передачу пакетов, передачу сообщений о получении пакетов и сброс пакетов по алгоритму RED.

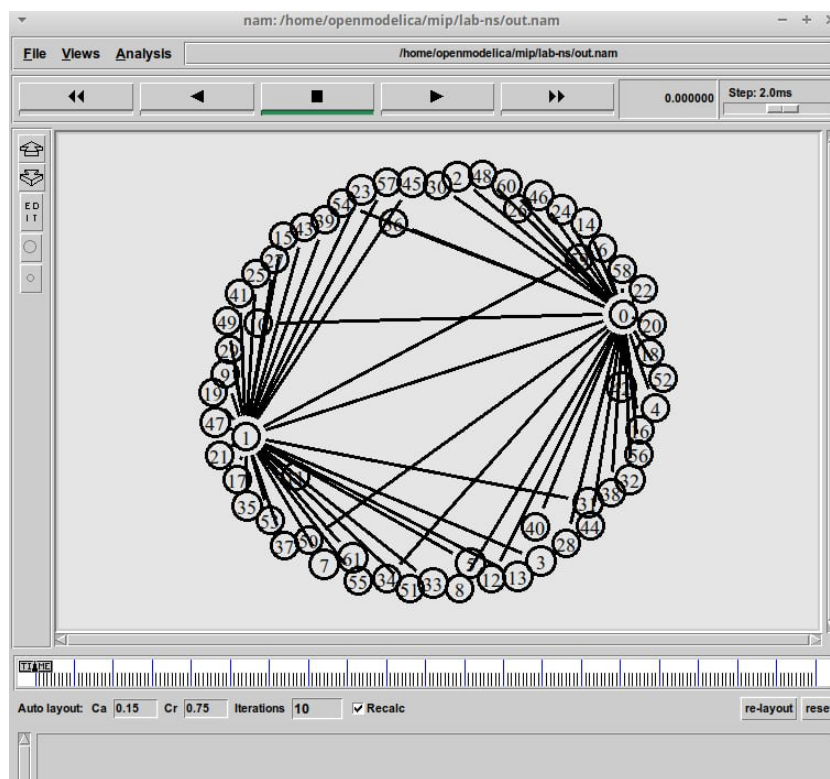


Рис. 3.1: Схема моделируемой сети при N=30

Далее выводятся графики:

1. График изменения размера окна TCP на всех источниках (xgraph)

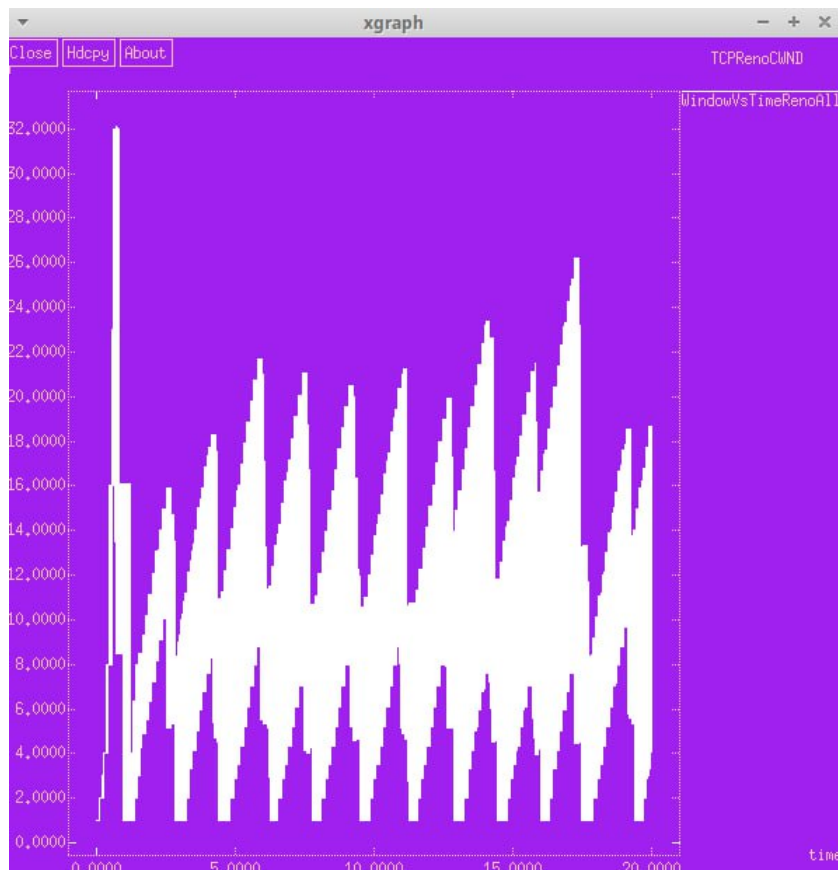


Рис. 3.2: Изменение размера окна TCP на всех источниках (xgraph)

2. Изменение размера окна TCP на линке 1-го источника(xgraph). Тут мы видим периодическое уменьшение и увеличение окна TCP. Что-то похожее было в прошлых лабораторных работах когда мы использовали TCP Reno.

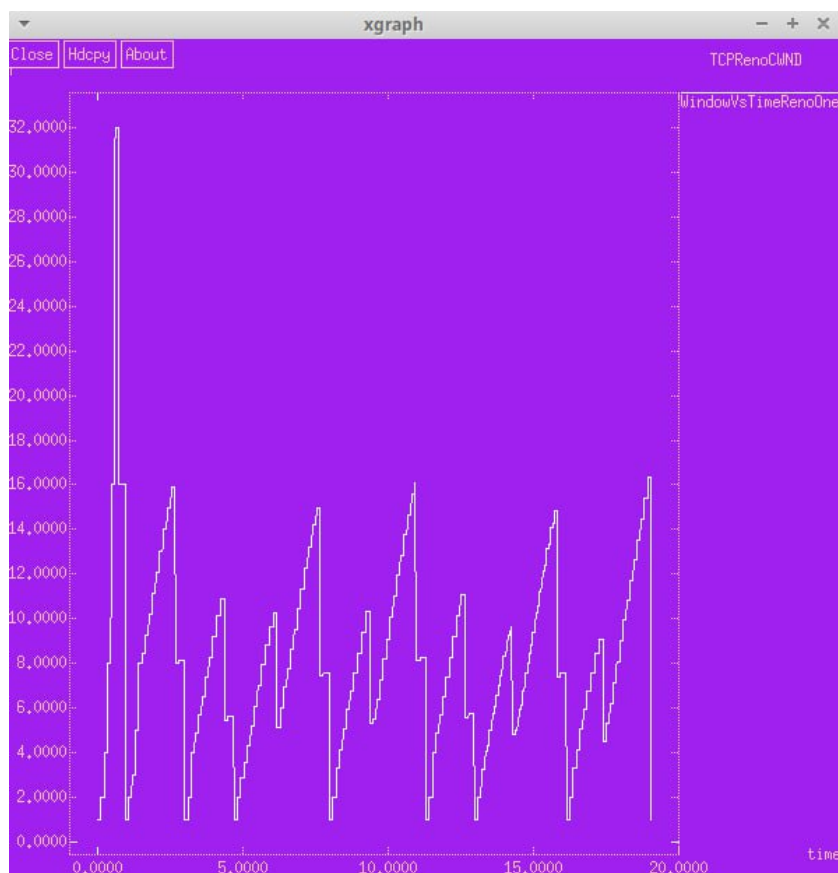


Рис. 3.3: Изменение размера окна TCP на линке 1-го источника(xgraph)

3. Изменение размера длины очереди на линке (R1–R2)(xgraph). В данном графике мы видим что после достижения максимального значения 110000(примерно) средняя длина очереди колеблется с одинаковой частотой, достигая значения не выше 80000

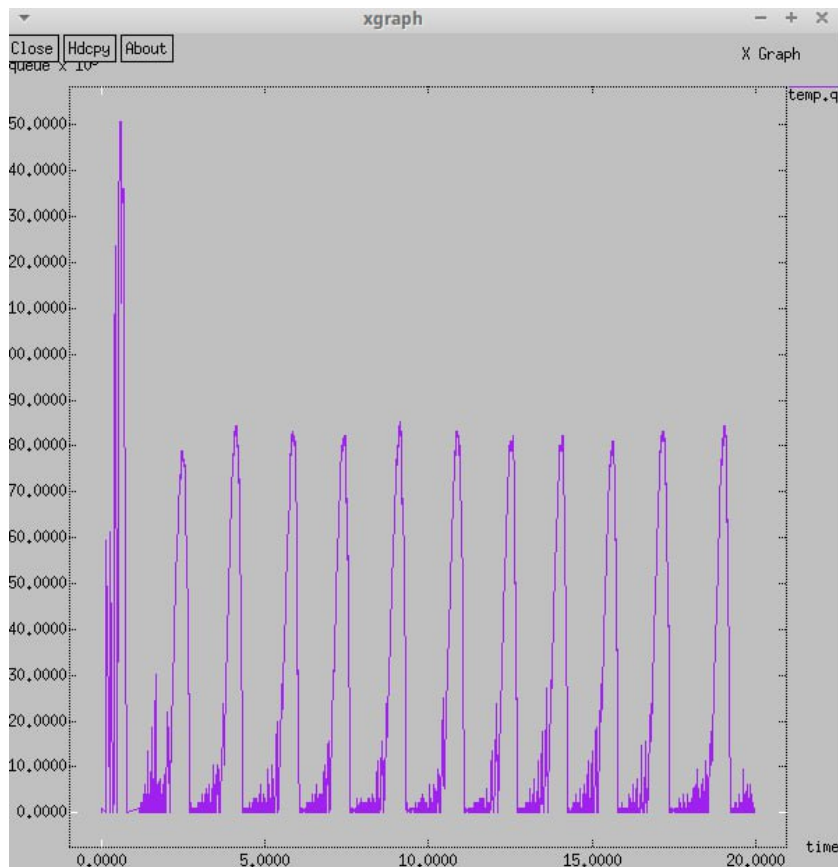


Рис. 3.4: Изменение размера длины очереди на линке (R1–R2)(xgraph)

4. Изменение размера средней длины очереди на линке (R1–R2)(xgraph). В данном графике мы видим максимальное значение, а затем идет колебание периодическое, схожее с прошлым графиком (3).

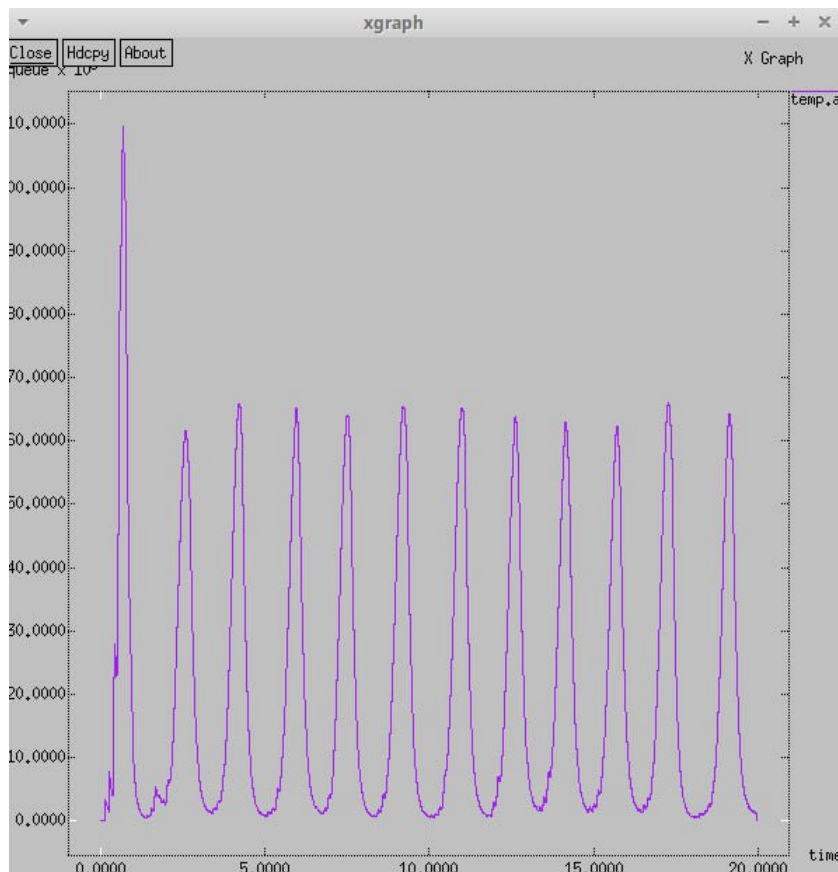


Рис. 3.5: Изменение размера средней длины очереди на линке (R1–R2)(xgraph)

После работы с xgraph переходим к GNU Plot и пишем программу для построения идентичных графиков в GNU Plot. Задаём им те же цвета, сохраняем их в пнг формате

```
#!/usr/bin/gnuplot -persist
# задаём текстовую кодировку,
# тип терминала, тип и размер шрифта

set encoding utf8
set term pngcairo font "Helvetica,9"

# задаём выходной файл графика
set out 'window_1.png'
```



```

# задаём название графика
set title "Изменение размера окна TCP на линке 1-го источника при N=30"

# подписи осей графика
set xlabel "t[s]" font "Helvetica, 10"
set ylabel "CWND [pkt]" font "Helvetica, 10"

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

# задаём выходной файл графика
set out 'window_2.png'

# задаём название графика
set title "Изменение размера окна TCP на всех N источниках при N=30"

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

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

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

# подписи осей графика
set xlabel "t[s]" font "Helvetica, 10"

```

```

set ylabel "Queue Length [pkt]" font "Helvetica, 10"

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

# задаём выходной файл графика
set out 'av_queue.png'

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

# подписи осей графика
set xlabel "t[s]" font "Helvetica, 10"
set ylabel "Queue Avg Length [pkt]" font "Helvetica, 10"

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

```

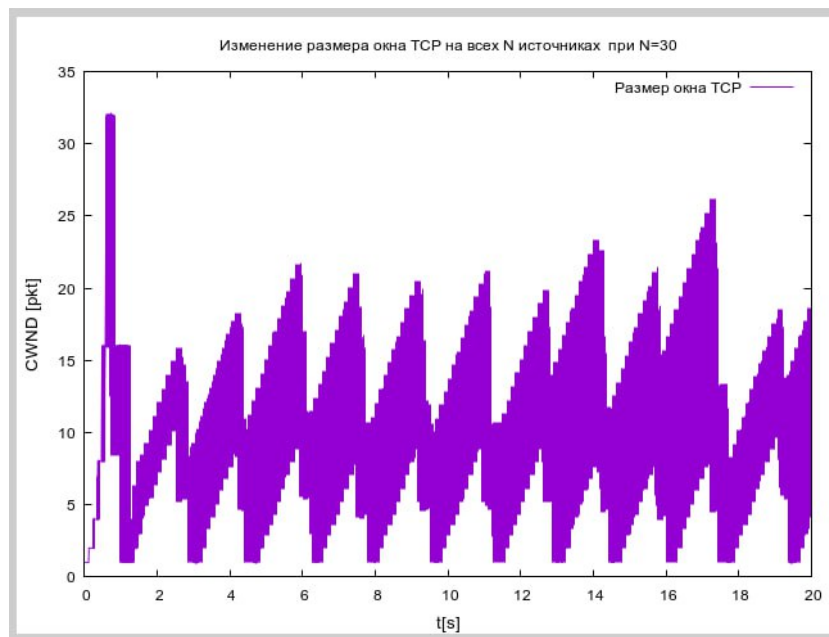


Рис. 3.6: Изменение размера окна TCP на всех источниках (GNU)

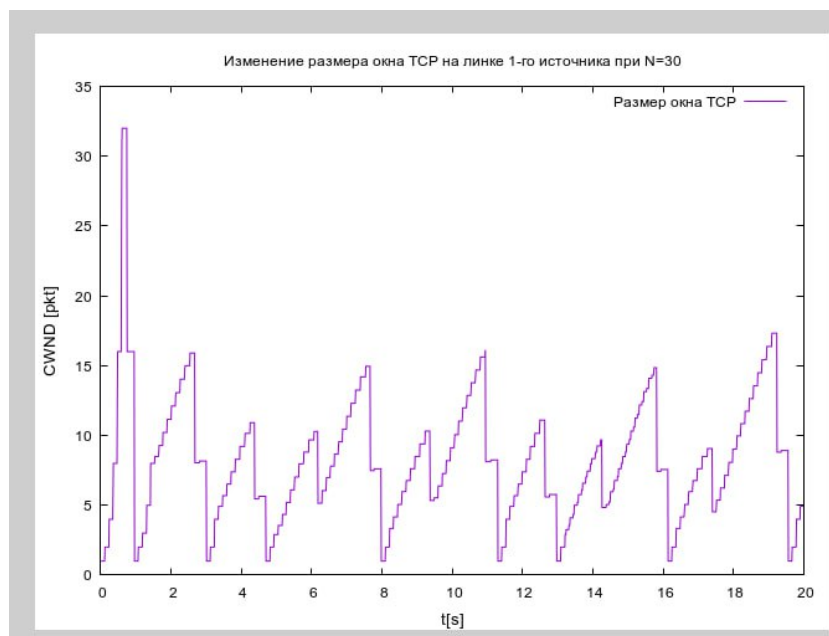


Рис. 3.7: Изменение размера окна TCP на линке 1-го источника (GNU)

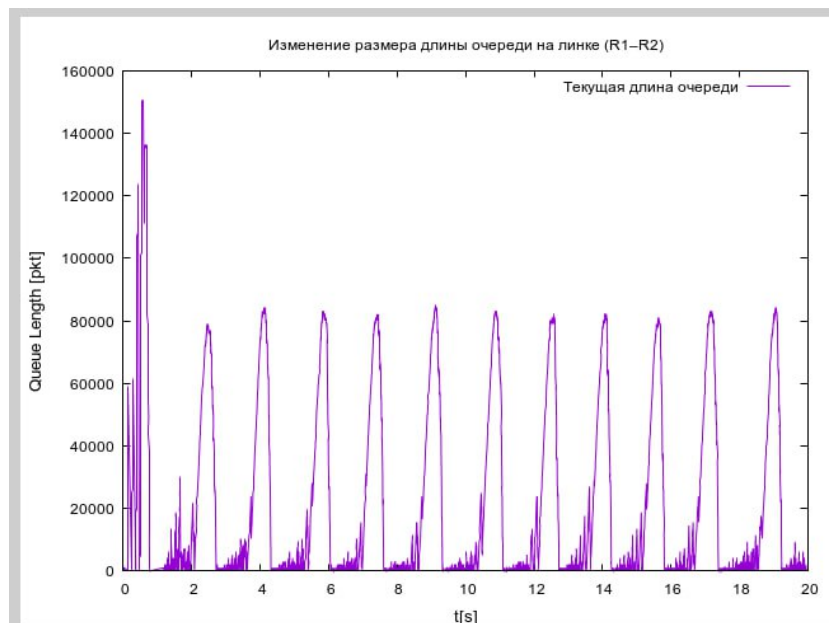


Рис. 3.8: Изменение размера длины очереди на линке (R1-R2)(GNU)

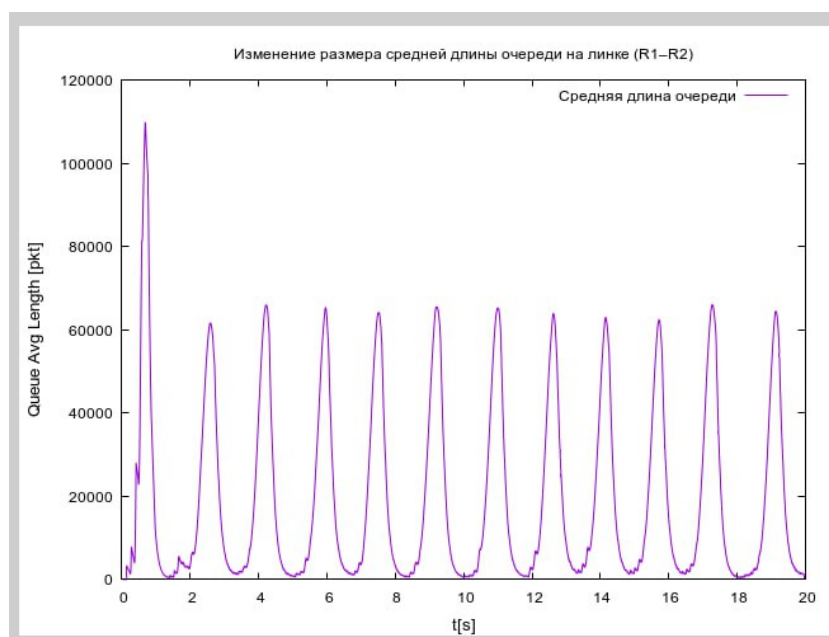


Рис. 3.9: Изменение размера средней длины очереди на линке (R1-R2)(GNU)

## 4 Выводы

При выполнении лабораторной работы я сделал задание для индивидуального выполнения.

## **Список литературы**

1. 1.5\_Королькова А. В., Кулябов Д. С. - Моделирование информационных процессов. Лабораторные работы.pdf