

Моделирование информационных процессов

Лабораторная работа №4

ПОДГОТОВИЛ:

Сергеев Тимофей

Сергеевич

ГРУППА НФИбд-02-20

СТ.Б: 1032201669

Цель работы

- Закрепить на практике изученную тему «Имитационное моделирование в NS-2», которую мы осваивали на протяжении предыдущих лабораторных работ.

Описание моделируемой сети

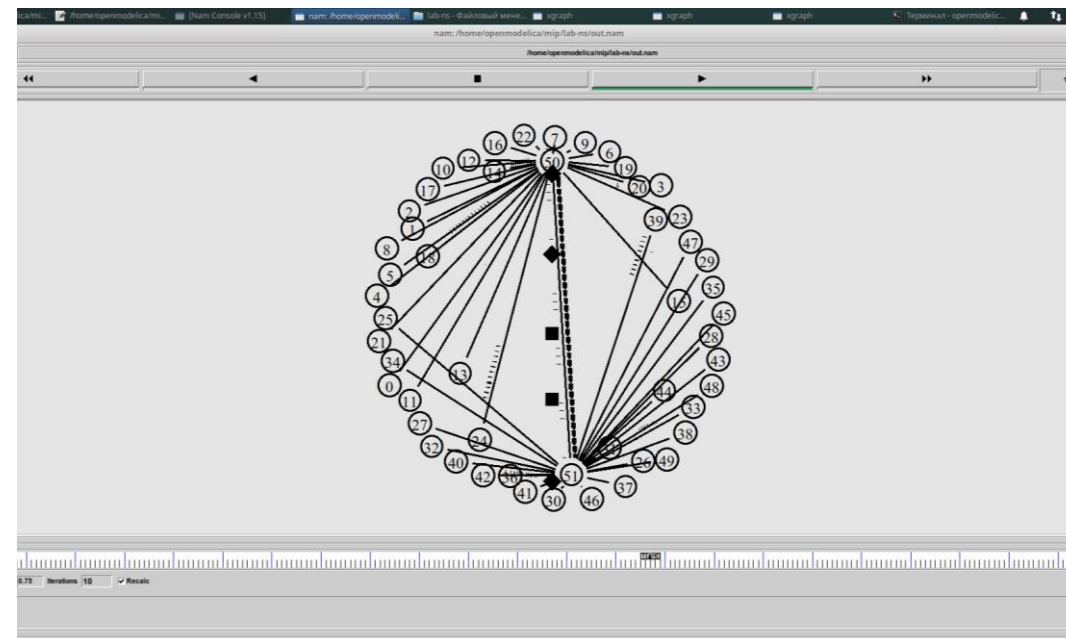
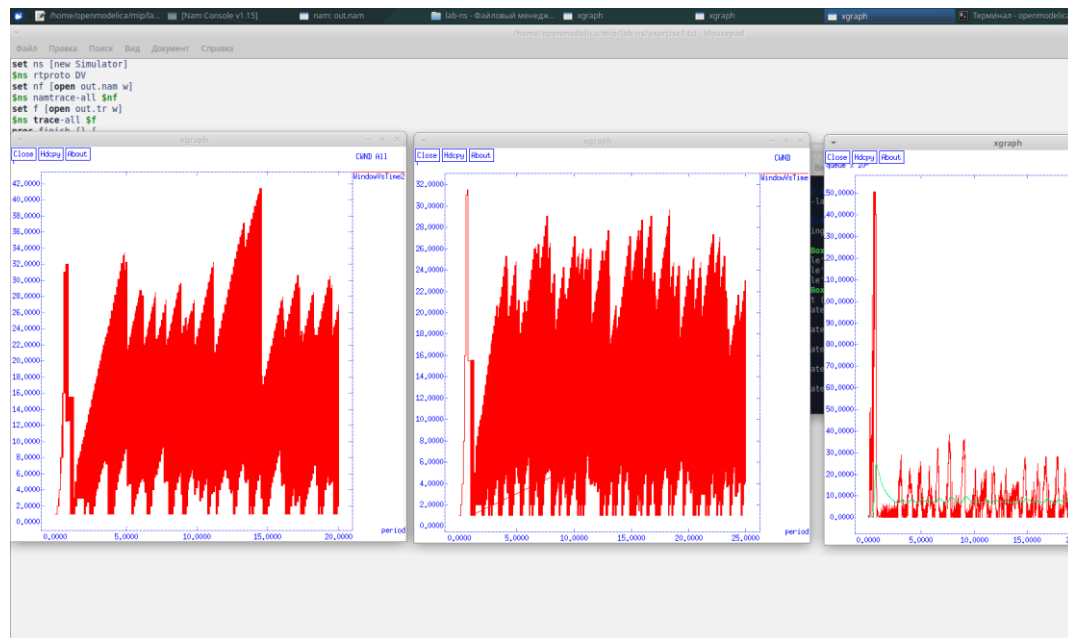
- – сеть состоит из N TCP-источников, N TCP-приёмников, двух маршрутизаторов $R1$ и $R2$ между источниками и приёмниками (N — не менее 20);
- – между TCP-источниками и первым маршрутизатором установлены дуплексные соединения с пропускной способностью 100 Мбит/с и задержкой 20 мс очередью типа DropTail;
- – 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 set ns [new Simulator]
2 set nf [open out.nam w]
3 $ns namtrace-all $nf
4 set f [open out.tr w]
5 $ns trace-all $f
6 set N 25
7 for {set i 0} {$i < [expr $N + 2]} {incr i} {
8     set node_($i) [$ns node]
9 }
10 set node_r1 [$ns node]
11 set node_r2 [$ns node]
12 for {set i 0} {$i < $N} {incr i} {
13     $ns duplex-link $node_($i) $node_r1 100Mb 20ms DropTail
14     $ns duplex-link $node_([expr $i + 25]) $node_r2 100Mb 20ms DropTail
15 }
16 $ns simplex-link $node_r1 $node_r2 20Mb 15ms RED
17 $ns queue-limit $node_r1 $node_r2 300
18 $ns simplex-link $node_r2 $node_r1 15Mb 20ms DropTail
19 for {set i 0} {$i < $N} {incr i} {
20     set tcp($i) [$ns create-connection TCP/Reno $node_($i) TCPSink $node_([expr $i + 25]) 1]
21 }
22 set thresh 75
23 set maxthresh 150
24 set q_weight 0.002
25 set linterm 0.1
26 for {set i 0} {$i < $N} {incr i} {
27     $tcp($i) set window 32
28     set ftp($i) [$tcp($i) attach-source FTP]
29     $ftp($i) set packetSize 500
30 }
31 set windowVsTime [open WindowVsTime w]
32 set windowVsTime2 [open WindowVsTime2 w]
33 set qmon [$ns monitor-queue $node_r1 $node_r2 [open qm.out w] 0.1];
34 [$ns link $node_r1 $node_r2] queue-sample-timeout;
35 set redq [$ns link $node_r1 $node_r2] queue]
36 set tchan [open all.q w]
37 $redq trace curq
38 $redq trace ave
39 $redq attach $tchan
40 proc plotWindow {tcpSource file} {
41     global ns
42     set time 0.01
43     set now [$ns now]
44     set cwnd [$tcpSource set cwnd_]
45     puts $file "$now $cwnd"
46     $ns at [expr $now+$time] "plotWindow $tcpSource $file"
47 }
```

```
48 proc finish {} {
49     global tchan
50     set awkCode {
51         {
52             if ($1 == "Q" && NF>2) {
53                 print $2, $3 >> "temp.q";
54                 set end $2
55             }
56             else if ($1 == "a" && NF>2)
57                 print $2, $3 >> "temp.a";
58         }
59     }
60     set f [open temp.queue w]
61     puts $f "TitleText: red"
62     puts $f "Device: Postscript"
63     if { [info exists tchan_] } {
64         close $tchan_
65     }
66     exec rm -f temp.q temp.a
67     exec touch temp.a temp.q
68     exec awk $awkCode all.q
69     puts $f "\nqueue"
70     exec cat temp.q >> $f
71     puts $f "\n\naverage queue_size"
72     exec cat temp.a >> $f
73     close $f
74     exec xgraph -bb -tk -bg white -fg blue 0.Style -x period -t "CWND" WindowVsTime &
75     exec xgraph -bb -tk -bg white -fg blue 0.Style -x period -t "CWND All" WindowVsTime2 &
76     exec xgraph -bb -tk -bg white -fg blue 0.Style -x period -y queue temp.queue &
77     exec nam out.nam &
78     exit 0
79 }
80 for {set i 0} {$i < $N} {incr i} {
81     $ns at 0.0 "$ftp($i) start"
82     $ns at 0.0 "plotWindow $tcp($i) $windowVsTime2"
83 }
84 $ns at 0.0 "plotWindow $tcp(0) $windowVsTime"
85 $ns at 25.0 "finish"
86 $ns run
87 }
```

Выполнение задания



Результат

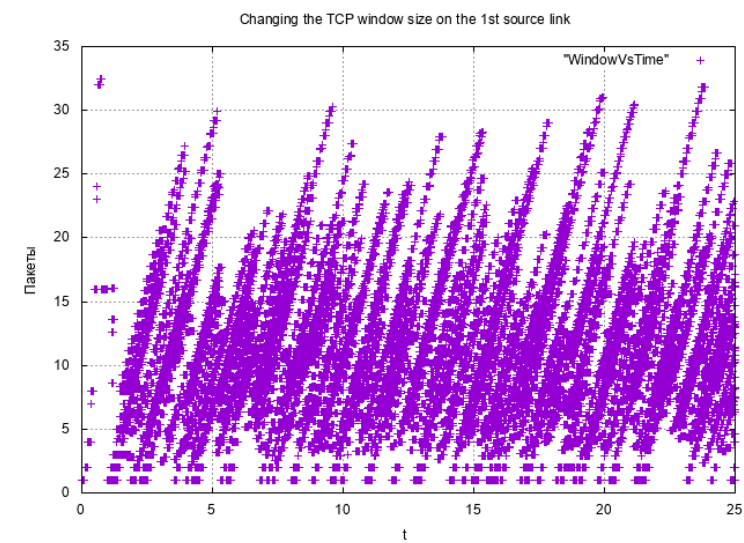
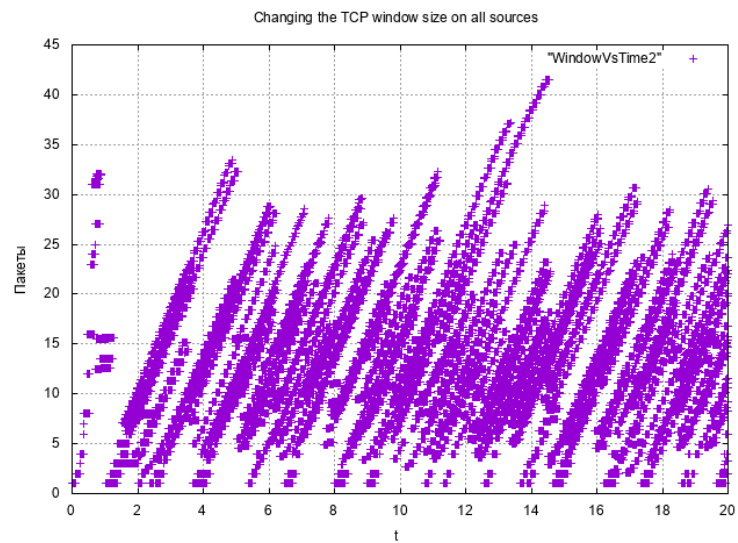
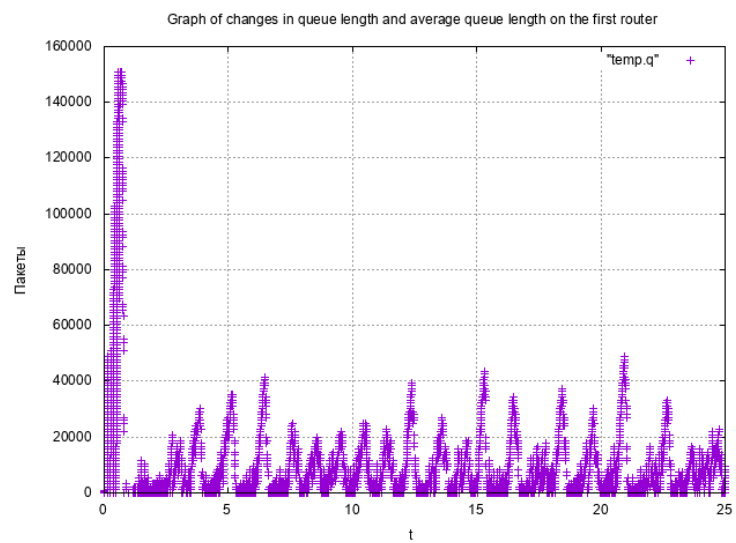
gnuplot

```
▼ /home/openmodelica
Файл  Правка  Поиск  Вид  Документ  Справка
#!/usr/bin/gnuplot -persist
set encoding utf8
set term png size 640, 480 enhanced font "Arial,9"
set grid
set style line 2

set output "1.png"
set title "Graph of changes in queue length and average queue length on the first router"
set xlabel "t"
set ylabel "Пакеты"
plot "temp.q"

set output "2.png"
set title "Changing the TCP window size on the 1st source link"
set xlabel "t"
set ylabel "Пакеты"
plot "WindowVsTime"

set output "3.png"
set title "Changing the TCP window size on all sources"
set xlabel "t"
set ylabel "Пакеты"
plot "WindowVsTime2"
```



Заключение

- В данной лабораторной работе мы смогли закрепить полученные ранее знания по работе с пакетом NS-2, Xgraph и GNUPlot, разработав имитационную модель самостоятельно.