

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

дисциплина: Моделирование информационных процессов

Студент:

Ibragimov Ulugbek

Группа:

НФИбд-02-20

МОСКВА

2023 г.

| | |
|--|----|
| Цель | 3 |
| Задачи | 4 |
| Выполнение лабораторной работы..... | 5 |
| Анализ результатов | 12 |
| Вывод..... | 13 |
| Приложение. Вид итоговых листингов. | 14 |

Цель

Приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

Задачи

1. Для приведённой схемы разработать имитационную модель в пакете NS-2 (рис. 1).
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.

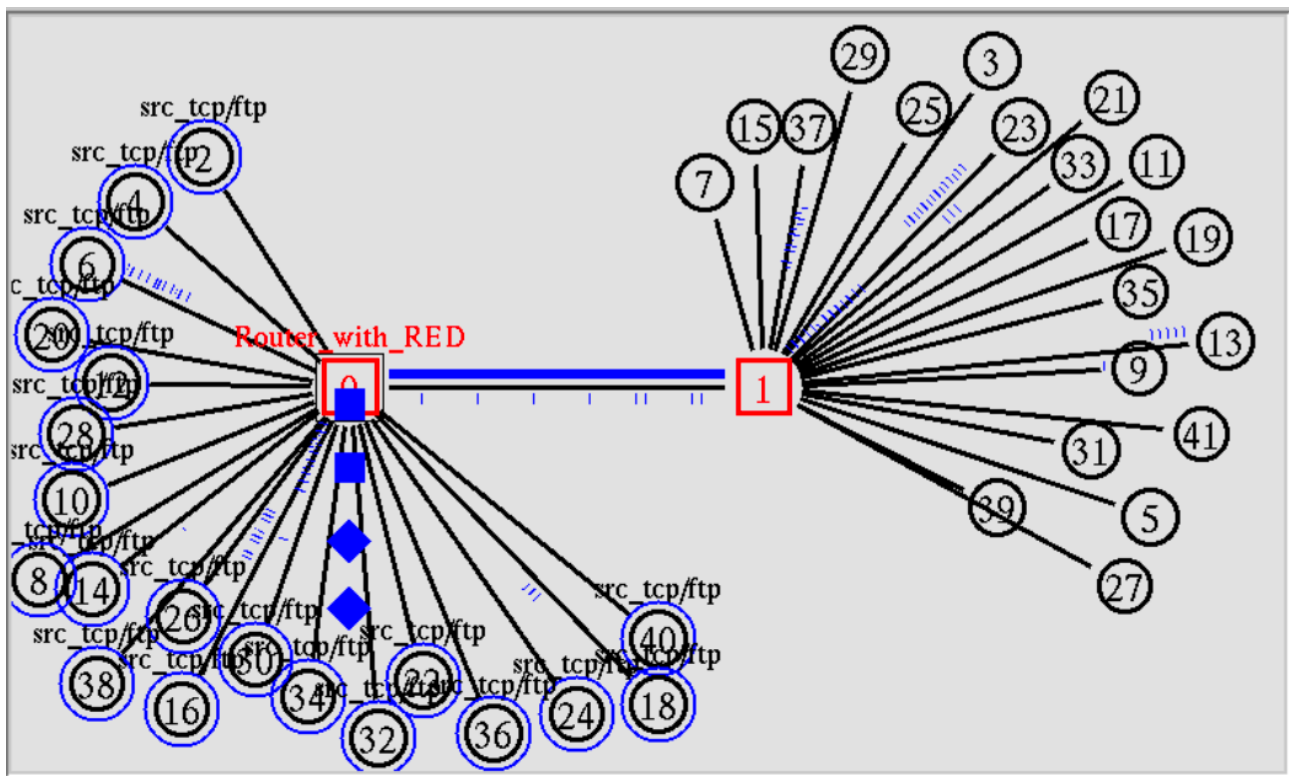


Рис.1. Схема моделируемой сети при N=20

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 TCPРeno;
- 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. Возьмем за основу, написанный шаблон. Перед строкой [`$ns at 5.0 «finish»`] будем писать модель. Сначала опишем сеть:

```
set N 20
set R1 [$ns node]
set R2 [$ns node]

$ns simplex-link $R2 $R1 15Mb 15ms DropTail
$ns simplex-link $R1 $R2 20Mb 15ms RED
$ns queue-limit $R1 $R2 300

set redq [[$ns link $R1 $R2] queue]
$redq set minthresh_ 75
$redq set maxthresh_ 150
$redq set q_weight_ 0.002
$redq set maxp_ 0.1

for {set i 0} {$i < $N} {incr i} {
    set node(src$i) [$ns node]
    $ns duplex-link $node(src$i) $R1 100Mb 20ms DropTail
    set node(dst$i) [$ns node]
    $ns duplex-link $node(dst$i) $R2 100Mb 20ms DropTail
```

```

    set tcp($i) [$ns create-connection TCP/Reno $node(src$i)
TCPSink $node(dst$i) $i]
    $tcp($i) set window_ 32
    $tcp($i) set packetSize_ 500
    set ftp($i) [$tcp($i) attach-source FTP]
    $ns at 0.0 "$ftp($i) start"
}

```

2. Далее изменим строку [`$ns at 5.0 «finish»`] на [`$ns at 20.0 «finish»`] и запустим скрипт (рис. 2-3):

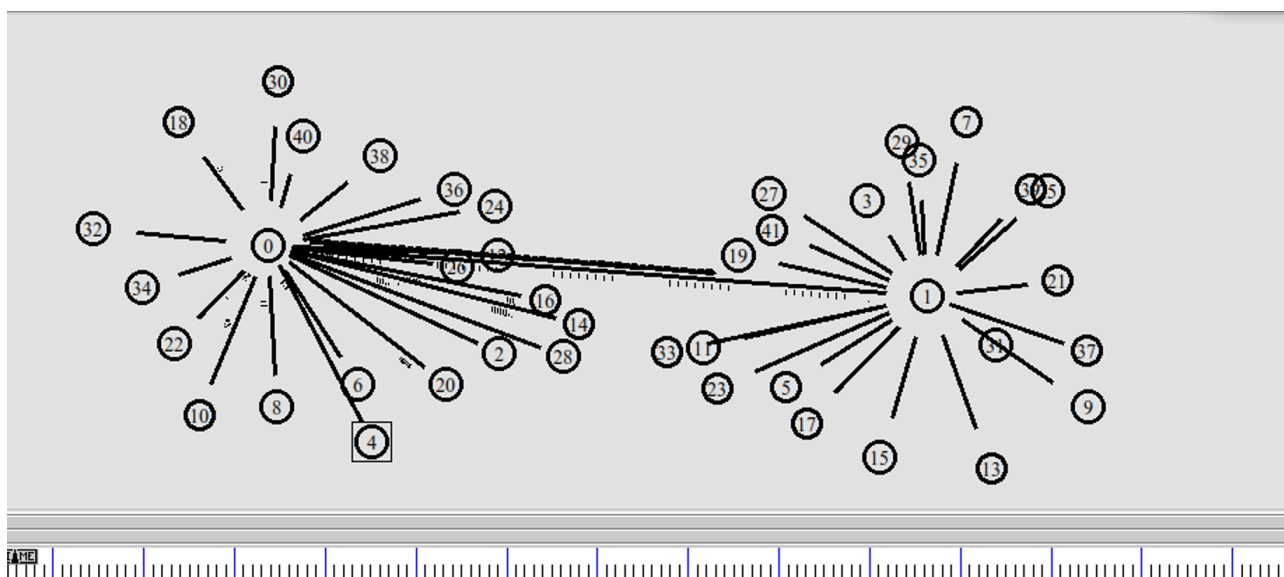


Рис.2. Запуск симуляции I.

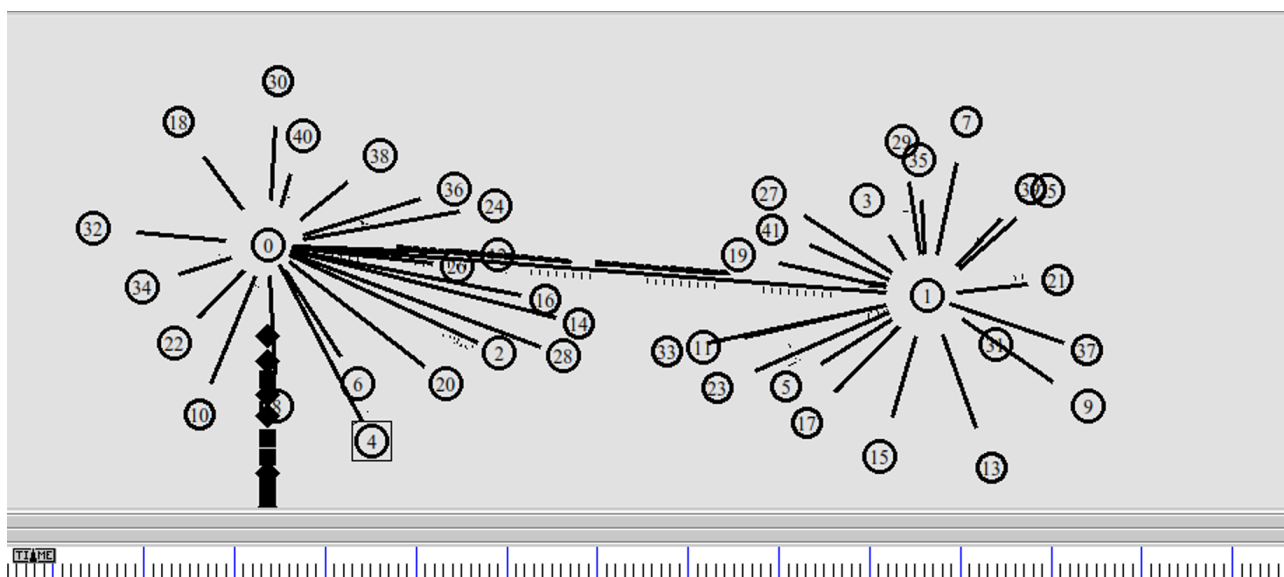


Рис.3. Запуск симуляции II.

3. Теперь приступим к построению графиков на xgraph. Для этого добавим в скрипт мониторинг размера окна одного источника и мониторинг размера всех источников:

```
set WindowVsTimeSRC0 [open WindowVsTimeSRC0 w]
set qmon [$ns monitor-queue $R1 $R2 [open qm.out w] 0.1];
[$ns link $R1 $R2] queue-sample-timeout;
$ns at 0.0 "plotWindow $tcp(0) $WindowVsTimeSRC0"

set WindowVsTimeSRC [open WindowVsTimeSRC w]
for {set i 0} {$i < $N} {incr i} {
    $ns at 0.0 "plotWindow $tcp($i) $WindowVsTimeSRC"
}
```

4. Также добавим мониторинг очереди:

```
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_
```

5. Добавим процедуру для формирования файла с данными о размере окна:

```
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"
}
```

6. Также изменим процедуру finish:

```
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";
        }
    }
    set f [open temp.queue w]
    puts $f "TitleText: red"
    puts $f "Device: Postscript"
    if { [info exists tchan_] } {
```

```

        close $tchan_
    }
    exec rm -f temp.q temp.a
    exec touch temp.a temp.q
    # выполнение кода AWK
    exec awk $awkCode all.q
    puts $f "\"queue
    exec cat temp.q >@ $f
    puts $f \"\\\"ave_queue
    exec cat temp.a >@ $f
    close $f
    # Запуск xgraph с графиками окна TCP и очереди:
    exec xgraph -bg white -x time -y WindowSize -bb -tk -t
    "WindowVsTimeSRC0" WindowVsTimeSRC0 &
    exec xgraph -bg white -x time -y WindowSize -bb -tk -t
    "WindowVsTimeSRC" WindowVsTimeSRC &
    exec xgraph -bg white -x time -y Queue -bb -tk tem-
    p.queue &

    exec nam out.nam &
    exit 0
}

```

7. Запустим скрипт (рис. 4-6):

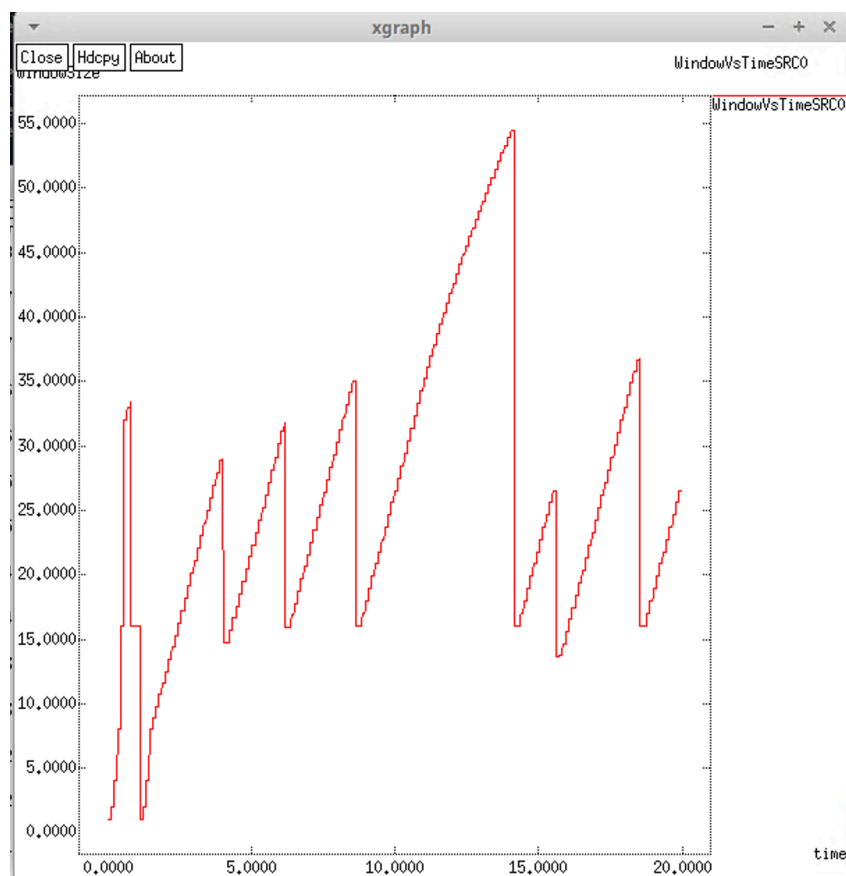


Рис. 4 Изменение размера окна TCP на линке 1-го источника (Xgraph)

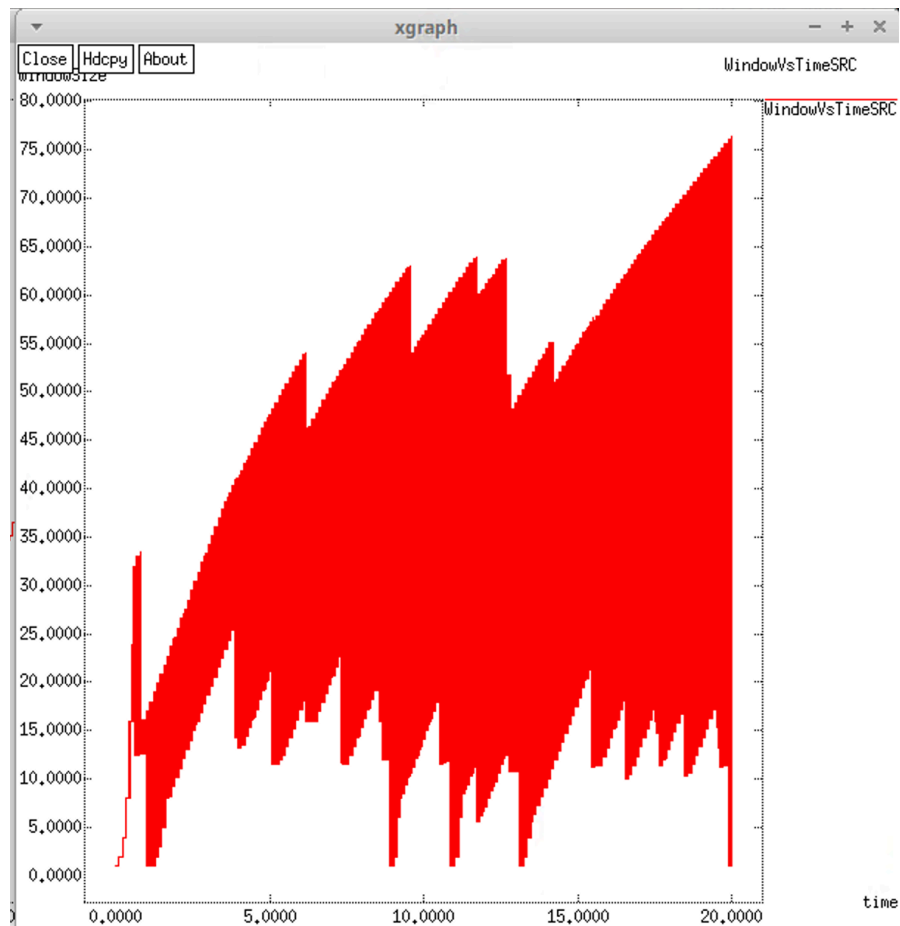


Рис. 5. Изменение размера окна TCP на всех источниках (Xgraph)

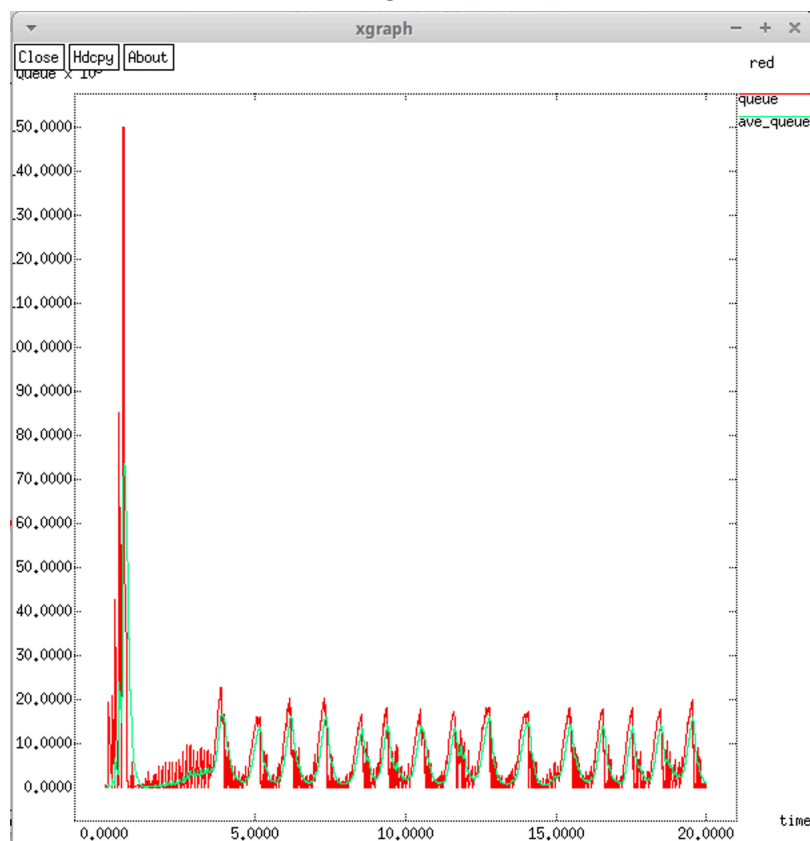


Рис. 6. Изменение размера длины очереди на линке (R1–R2) (Xgraph)

8. Теперь приступим к построению графиков на GNUplot. Для этого напишем следующий скрипт в новом файле (в той же директории):

```
#!/usr/bin/gnuplot -persist
set encoding utf8
set term pdfcairo font "Arial,9"
set out 'qm.pdf'
set style line 2

set title "WindowVsTimeSRC0"
set xlabel "time"
set ylabel "WindowSize"
plot "WindowVsTimeSRC0" using ($1):($2) with lines title "Queue"

set title "WindowVsTimeSRC"
set xlabel "time"
set ylabel "WindowSize"
plot "WindowVsTimeSRC" using ($1):($2) with lines title "Queue"

set title "Queue"
set xlabel "time"
set ylabel "queue"
plot "temp.q" using ($1):($2) with lines title "Queue", \
      "temp.a" using ($1):($2) with lines title «AvgQueue"
```

9. Далее сделаем этот файл исполняемым и запустим. На выходе имеем следующий файл pdf:

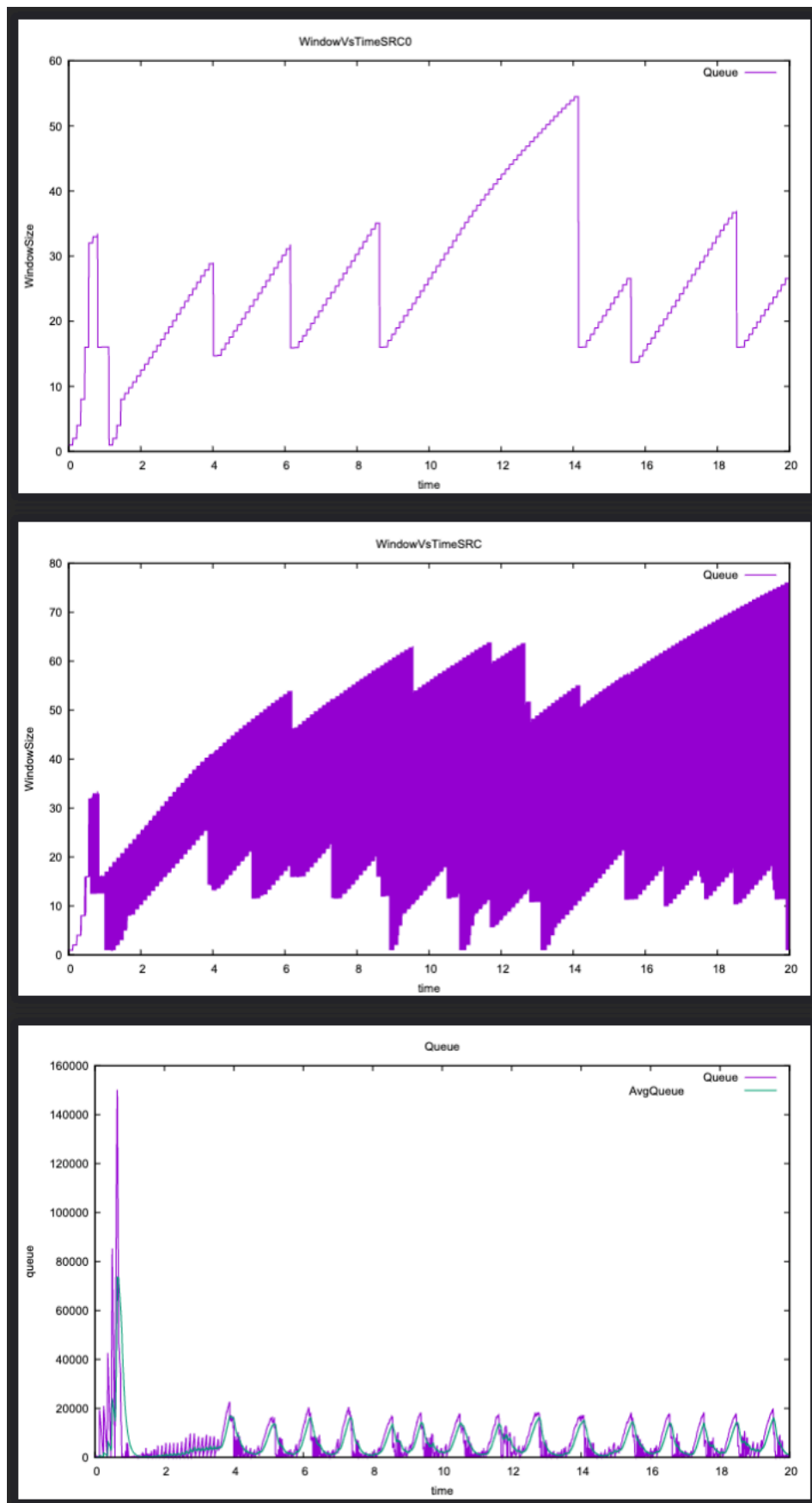


Рис. 7. Графики построенные при помощи GnuPlot.

Анализ результатов

В рамках выполнения работы произошли некоторые незначительные проблемы, но все они решены. В остальном работа выполнена без непредвиденных проблем в соответствии с руководством: ошибок и сбоев не произошло.

Стоит отметить, что написание скриптов NS-2 происходи крайне комфортно, синтаксис, объекты интуитивно понятны. Также сами скрипты довольно быстро выполняются.

Среда xgraph и GNUPlot также оказались удобными для использования. Но для реализации поставленных, задач необходимо было использовать документацию и руководство текущего курса.

Вывод

В результате выполнения работы, были улучшены практические навыки моделирования на NS-2. В добавок были улучшены навыки построения графиков при помощи средства GNUplot и Xgraph.

Приложение. Вид итоговых листингов.

model.tcl:

```
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 $f
6
7  proc plotWindow {tcpSource file} {
8      global ns
9      set time 0.01
10     set now [$ns now]
11     set cwnd [$tcpSource set cwnd_]
12     puts $file "$now $cwnd"
13     $ns at [expr $now+$time] "plotWindow $tcpSource $file"
14 }
15
16 proc finish {} {
17     global tchan_
18     set awkCode {
19         {
20             if ($1 == "Q" && NF>2) {
21                 print $2, $3 >> "temp.q";
22                 set end $2
23             }
24             else if ($1 == "a" && NF>2)
25                 print $2, $3 >> "temp.a";
26         }
27     }
28     set f [open temp.queue w]
29     puts $f "TitleText: red"
30     puts $f "Device: Postscript"
31     if { [info exists tchan_] } {
32         close $tchan_
33     }
34     exec rm -f temp.q temp.a
35     exec touch temp.a temp.q
36     # выполнение кода AWK
37     exec awk $awkCode all.q
38     puts $f "\"queue
39     exec cat temp.q >@ $f
40     puts $f "\\n\\\"ave_queue
41     exec cat temp.a >@ $f
42     close $f
43     # Запуск xgraph с графиками окна TCP и очереди:
44     exec xgraph -bg white -x time -y WindowSize -bb -tk -t "WindowVsTimeSRC0" WindowVsTimeSRC0 &
45     exec xgraph -bg white -x time -y WindowSize -bb -tk -t "WindowVsTimeSRC" WindowVsTimeSRC &
46     exec xgraph -bg white -x time -y Queue -bb -tk temp.queue &
47
48     exec nam out.nam &
49     exit 0
50 }
```

```

51
52 set N 20
53 set R1 [$ns node]
54 set R2 [$ns node]
55
56 $ns simplex-link $R2 $R1 15Mb 15ms DropTail
57 $ns simplex-link $R1 $R2 20Mb 15ms RED
58 $ns queue-limit $R1 $R2 300
59
60 set redq [$ns link $R1 $R2] queue
61 $redq set minthresh_ 75
62 $redq set maxthresh_ 150
63 $redq set q_weight_ 0.002
64 $redq set maxp_ 0.1
65
66 for {set i 0} {$i < $N} {incr i} {
67     set node(src$i) [$ns node]
68     $ns duplex-link $node(src$i) $R1 100Mb 20ms DropTail
69     set node(dst$i) [$ns node]
70     $ns duplex-link $node(dst$i) $R2 100Mb 20ms DropTail
71     set tcp($i) [$ns create-connection TCP/Reno $node(src$i) TCPSink $node(dst$i) $i]
72     $tcp($i) set window_ 32
73     $tcp($i) set packetSize_ 500
74     set ftp($i) [$tcp($i) attach-source FTP]
75     $ns at 0.0 "$ftp($i) start"
76 }
77
78 set tchan_ [open all.q w]
79 $redq trace curq_
80 $redq trace ave_
81 $redq attach $tchan_
82
83 set WindowVsTimeSRC0 [open WindowVsTimeSRC0 w]
84 set qmon [$ns monitor-queue $R1 $R2 [open qm.out w] 0.1];
85 [$ns link $R1 $R2] queue-sample-timeout;
86 $ns at 0.0 "plotWindow $tcp(0) $WindowVsTimeSRC0"
87
88 set WindowVsTimeSRC [open WindowVsTimeSRC w]
89 for {set i 0} {$i < $N} {incr i} {
90     $ns at 0.0 "plotWindow $tcp($i) $WindowVsTimeSRC"
91 }
92
93
94 $ns at 20.0 "finish"
95 $ns run

```

graph_plot

```
1  #!/usr/bin/gnuplot -persist
2  set encoding utf8
3  set term pdfcairo font "Arial,9"
4  set out 'qm.pdf'
5  set style line 2
6
7  set title "WindowVsTimeSRC0"
8  set xlabel "time"
9  set ylabel "WindowSize"
10 plot "WindowVsTimeSRC0" using ($1):($2) with lines title "Queue"
11
12 set title "WindowVsTimeSRC"
13 set xlabel "time"
14 set ylabel "WindowSize"
15 plot "WindowVsTimeSRC" using ($1):($2) with lines title "Queue"
16
17 set title "Queue"
18 set xlabel "time"
19 set ylabel "queue"
20 plot "temp.q" using ($1):($2) with lines title "Queue", \
21      "temp.a" using ($1):($2) with lines title "AvgQueue"
```