Лабораторная работа № 4

Задание для самостоятельного выполнения

Демидова Екатерина Алексеевна

Содержание

# 1 Цель работы

Разработать имитационную модель в пакете NS-2 и построить график изменения размера окна и TCP и график изменения длины очереди и средней длины очереди на первом маршрутизаторе (в Xgraph и в GNUPlot);

# 2 Описание моделируемой сети

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 TCPReno;
* параметры алгоритма RED: qmin = 75, qmax = 150, qw = 0, 002, pmax = 0.1;
* максимальный размер TCP-окна 32; размер передаваемого пакета 500 байт; время моделирования – не менее 20 единиц модельного времени.

# 3 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.

# 4 Выполнение лабораторной работы

Создадим сеть. Зададим N = 25 TCP-источников, N = 25 TCP-приёмников, два маршрутизатора r1 и r2 между источниками и приёмниками. Между TCP-источниками и первым маршрутизатором установим 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 TCPReno. Параметры алгоритма RED: qmin = 75, qmax = 150, qw = 0, 002, pmax = 0.1 задаются соответствено следующим образом:

# 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

* thresh\_ – минимальная граница для среднего размера очереди в пакетах;
* maxthresh\_ – максимальная граница для среднего размера очереди в пакетах;
* q\_weight\_ – вес очереди, используется для вычисления среднего размера очереди;
* linterm\_– как средний размер очереди варьируется между thresh\_ и maxthresh\_, так и вероятность отбрасывания пакета варьируется между 0 и 1/linterm\_;

Также задаем максимальный размер TCP-окна 32; размер передаваемого пакета 500 байт; время моделирования – 25 единиц модельного времени. Ниже приведён листинг реализации описанной модели:

#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) $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

Сеть имеет вид(рис. [??]).

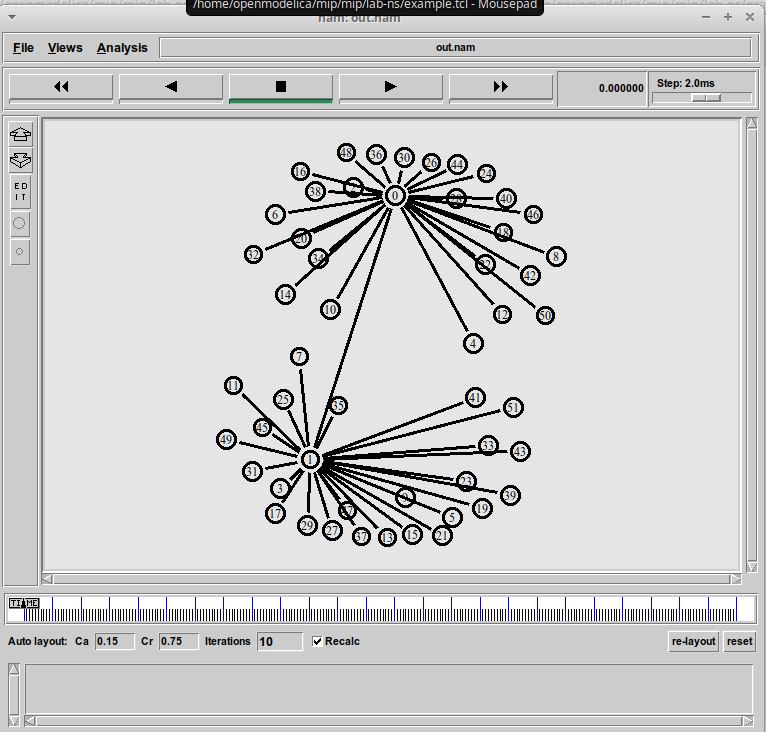
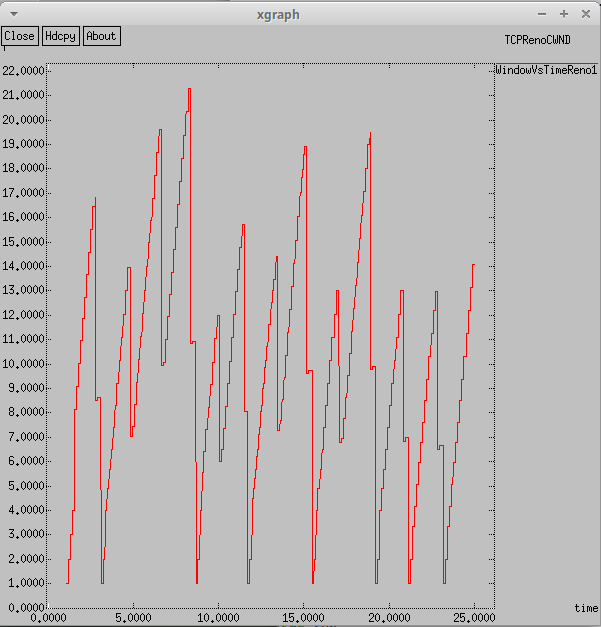
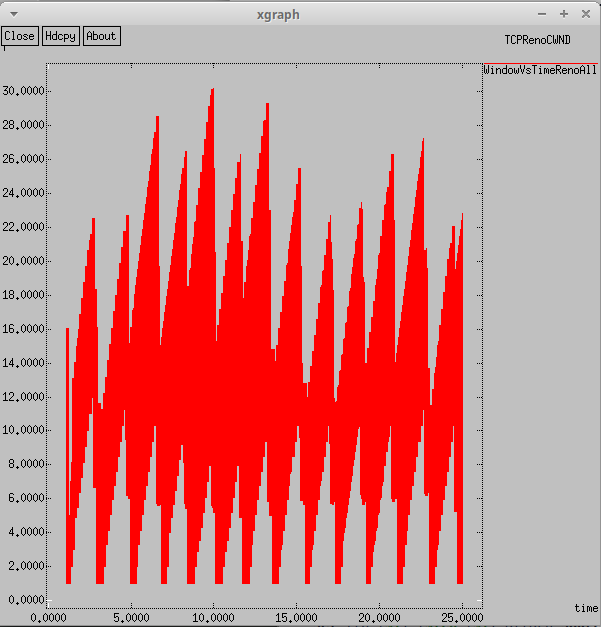


Схема сети

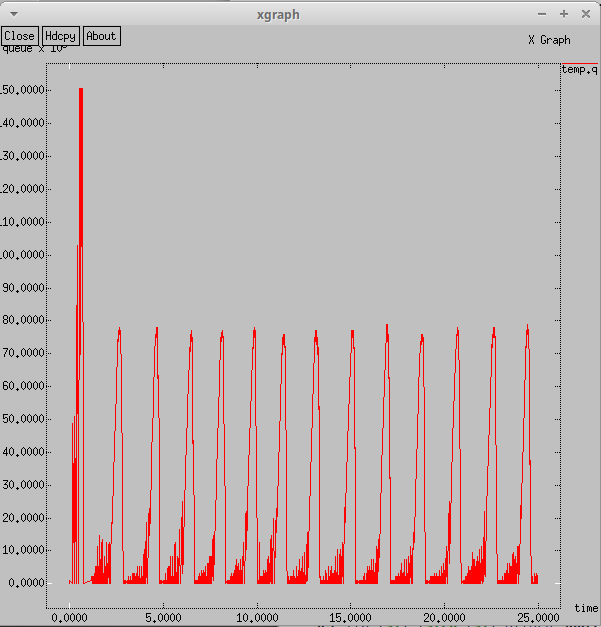
В результате получим следующие графики(рис. [??] - [??]).



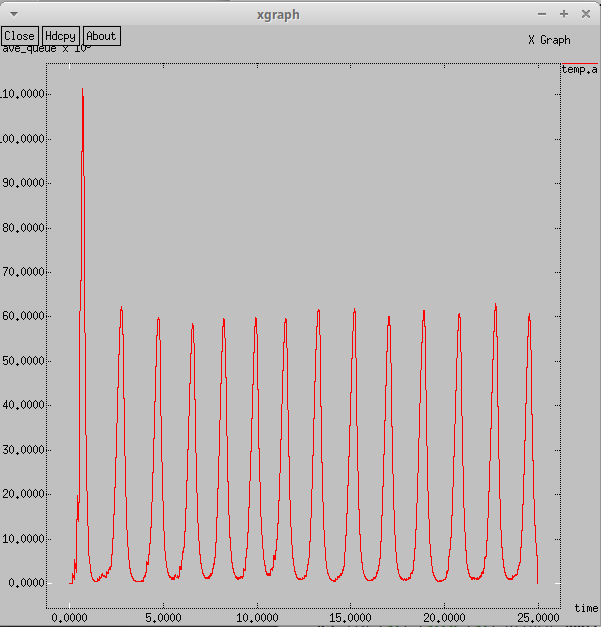
Изменение размера окна TCP на линке 1-го источника при N=25. Xgraph



Изменение размера окна TCP на всех источника при N=25. Xgraph



Изменение размера длины очереди на линке (R1–R2) при N=25, qmin = 75, qmax = 150. Xgraph

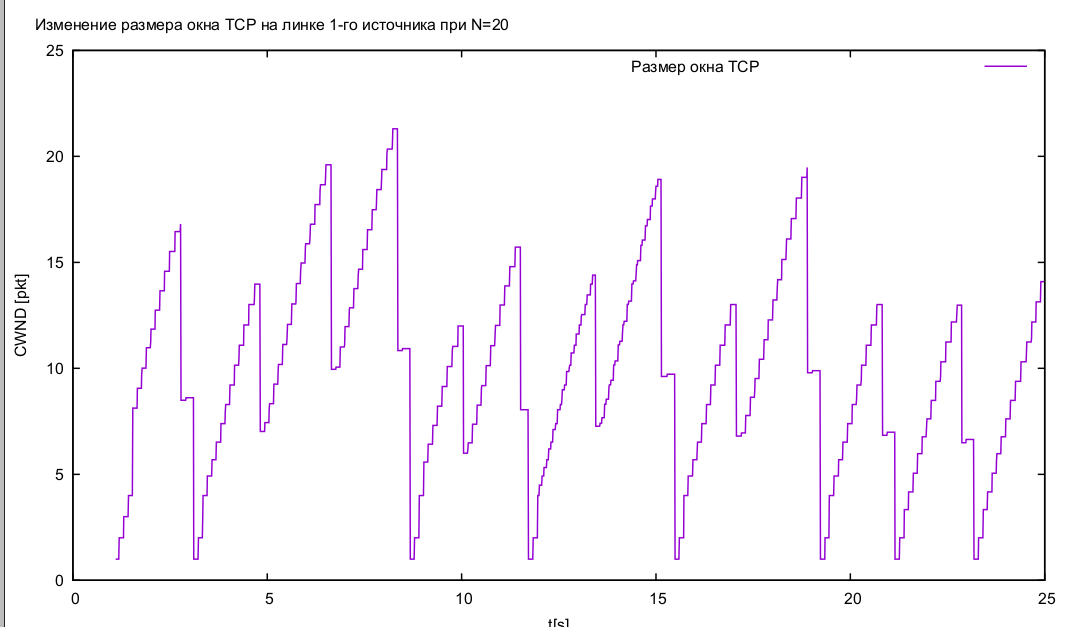


Изменение размера средней длины очереди на линке (R1–R2) при N=25, qmin = 75, qmax = 150. Xgraph

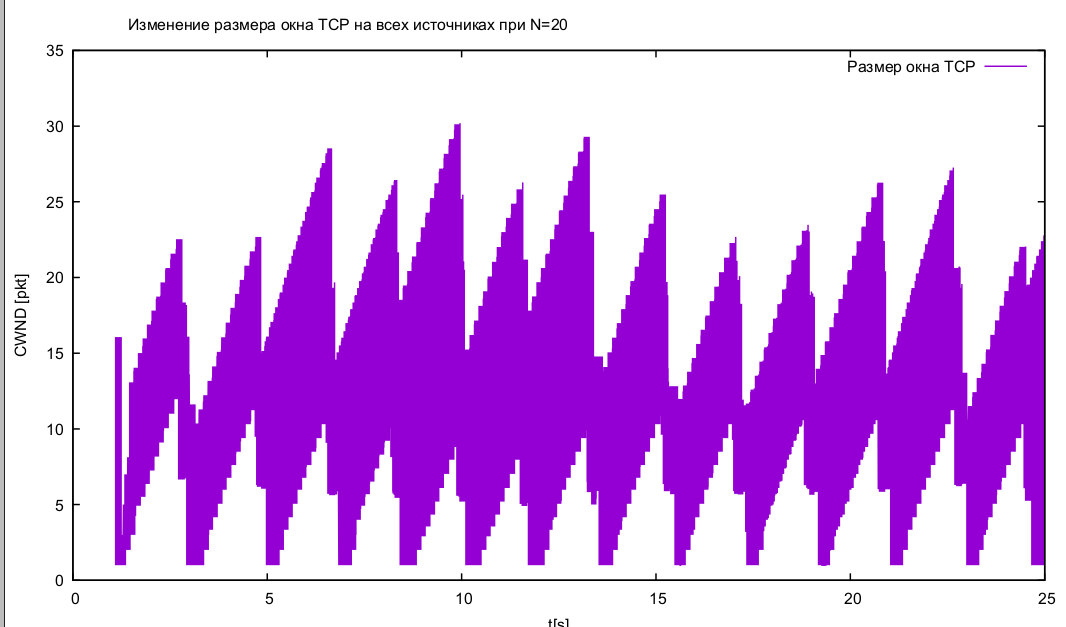
Теперь нарисуем график поведения длины очереди с помощью graph\_plot. Ниже приведен листинг для отрисовки:

#!/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, qmax = 150"  
# 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, qmin = 75, qmax = 150"  
# 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"

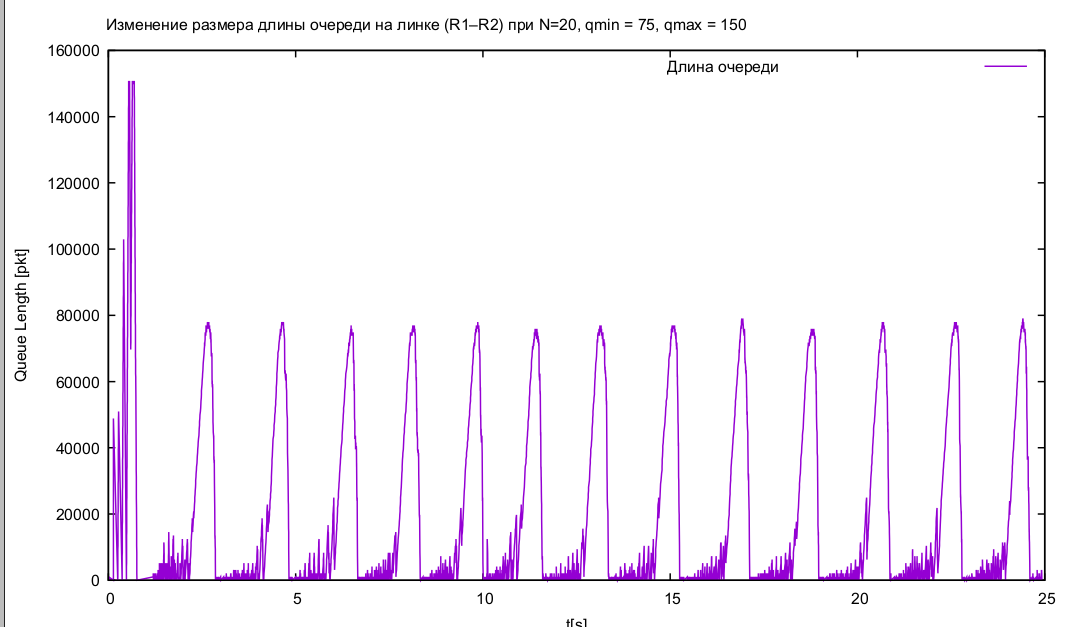
Сделаем этот файл исполняемым и запустим скрипт, который создаст pdf-файлы с результатами моделирования(рис. [??] - [??]).



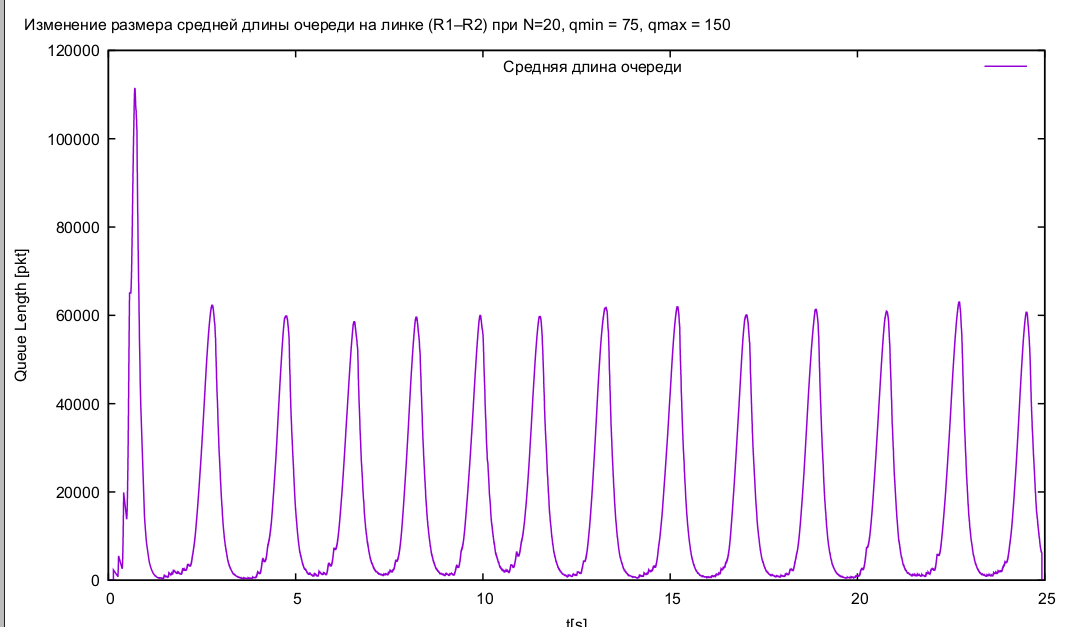
Изменение размера окна TCP на линке 1-го источника при N=25. GNUPlot



Изменение размера окна TCP на всех источника при N=25. GNUPlot



Изменение размера длины очереди на линке (R1–R2) при N=25, qmin = 75, qmax = 150. GNUPlot



Изменение размера средней длины очереди на линке (R1–R2) при N=25, qmin = 75, qmax = 150. GNUPlot

# 5 Выводы

В результате выполнения работы была разработана имитационная модель в пакете NS-2 и построены график изменения размера окна и TCP и график изменения длины очереди и средней длины очереди на первом маршрутизаторе (в Xgraph и в GNUPlot).