Лабораторная работа №4

Задание для самостоятельного выполнения

Игнатенкова Варвара Николаевна

Содержание

[1 Цель работы 1](#_Toc192296736)

[2 Задание 1](#_Toc192296737)

[3 Выполнение лабораторной работы 1](#_Toc192296738)

[4 Выводы 10](#_Toc192296739)

# 1 Цель работы

Выполнить задание для самостоятельного выполнения.

# 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 TCPReno;
* параметры алгоритма RED: ;
* максимальный размер TCP-окна 32; размер передаваемого пакета 500 байт; время моделирования — не менее 20 единиц модельного времени.

Откроем файл .tcl на редактирование, в нем построим сеть. Зададим N = 20 TCP-источников, N = 20 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. Зададим также параметры алгоритма RED: qmin = 75, qmax = 150, qw = 0, 002, pmax = 0.1. Также нам нужно выполнить мониторинг окна TCP и мониторинг очереди. Листинг такой программы выглядит следующим образом:

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

set f [open out.tr w]

$ns trace-all $f

Agent/TCP set window\_ 32

Agent/TCP set pktSize\_ 500

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";

}

}

exec rm -f temp.q temp.a

exec touch temp.a temp.q

exec awk $awkCode all.q

exec xgraph -fg green -bg blue -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeRenoOne &

exec xgraph -fg green -bg blue -bb -tk -x time -t "TCPRenoCWND" 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

}

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 20

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]

}

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;

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"

$ns at 20.0 "finish"

$ns runЗапустив созданную программу на выполнение получим nam файл со схемой моделируемой сети (рис. 1).

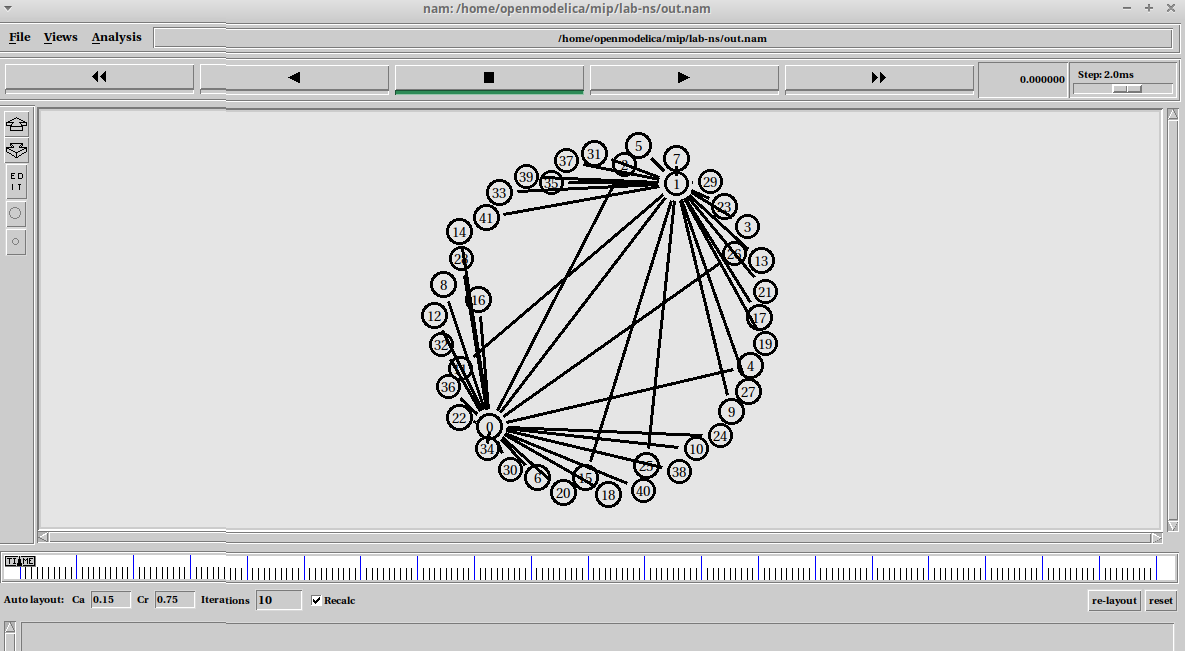


Рис. 1: Схема моделируемой сети при N=20

Также получим графики изменения размера окна TCP на линке 1-го источника (рис. 2) и на всех источниках (рис. 3). Графики построены с помощью xgraph.

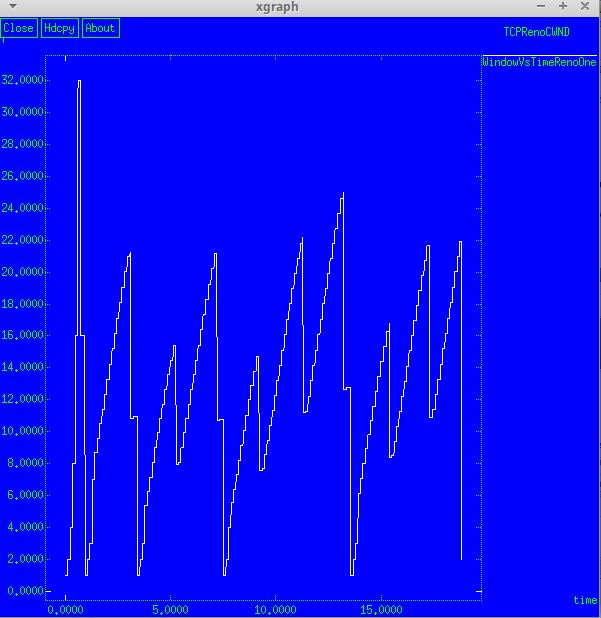


Рис. 2: Изменение размера окна TCP на линке 1-го источника при N=20

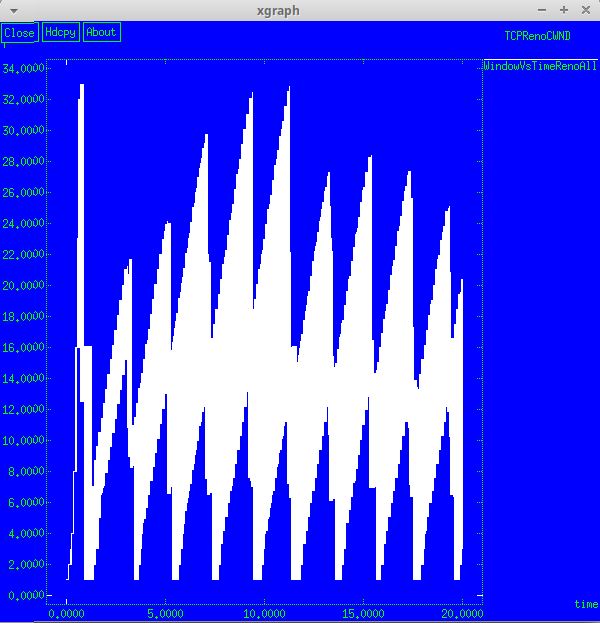


Рис. 3: Изменение размера окна TCP на всех источниках при N=30

Еще получим графики изменения размера длины очереди (рис. 4) и размера средней длины очереди (рис. 5). Графики построены с помощью xgraph.

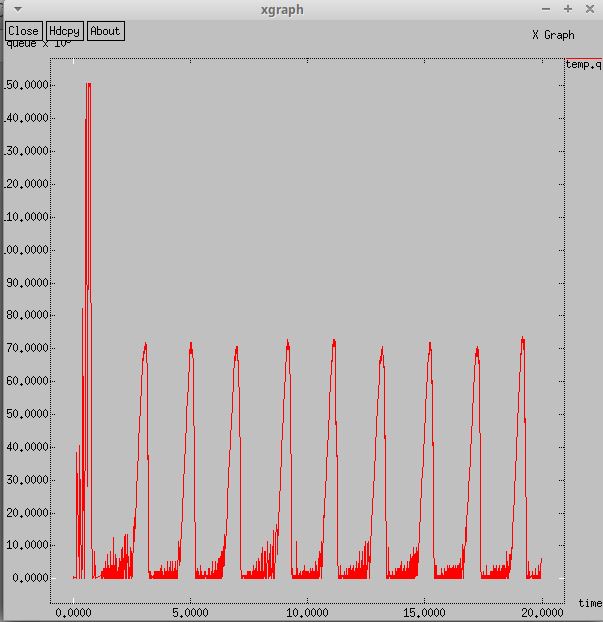


Рис. 4: Изменение размера длины очереди на линке (R1–R2) при N=30

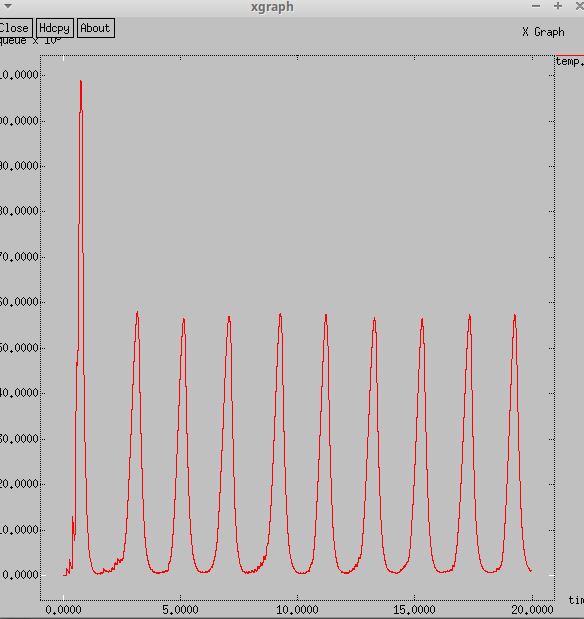


Рис. 5: Изменение размера средней длины очереди на линке (R1–R2) при N=30

Напишем программу для построения графиков в GNUPlot:

#!/usr/bin/gnuplot -persist

set encoding utf8

set term pngcairo font "Helvetica,9"

set out 'window\_1.png'

set title "Изменение размера окна TCP на линке 1-го источника при N=20"

set xlabel "t[s]" font "Helvetica, 10"

set ylabel "CWND [pkt]" font "Helvetica, 10"

plot "WindowVsTimeRenoOne" using ($1):($2) with lines title "Размер окна TCP"

set out 'window\_2.png'

set title "Изменение размера окна TCP на всех N источниках при N=20"

plot "WindowVsTimeRenoAll" using ($1):($2) with lines title "Размер окна TCP"

set out 'queue.png'

set title "Изменение размера длины очереди на линке (R1–R2)"

set xlabel "t[s]" font "Helvetica, 10"

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

plot "temp.q" using ($1):($2) with lines title "Текущая длина очереди"

set out 'av\_queue.png'

set title "Изменение размера средней длины очереди на линке (R1–R2)"

set xlabel "t[s]" font "Helvetica, 10"

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

plot "temp.a" using ($1):($2) with lines title "Средняя длина очереди"Сделаем исполняемым и запустим его. Получим 4 графика.

Графики изменения размера окна TCP на линке 1-го источника (рис. 6) и на всех источниках (рис. 7).

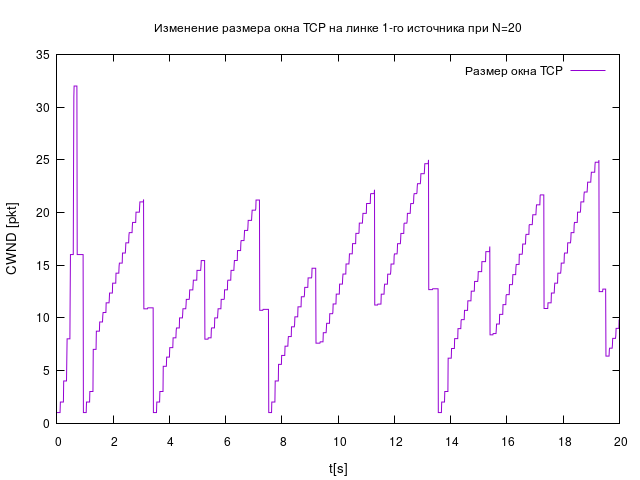


Рис. 6: Изменение размера окна TCP на линке 1-го источника при N=20

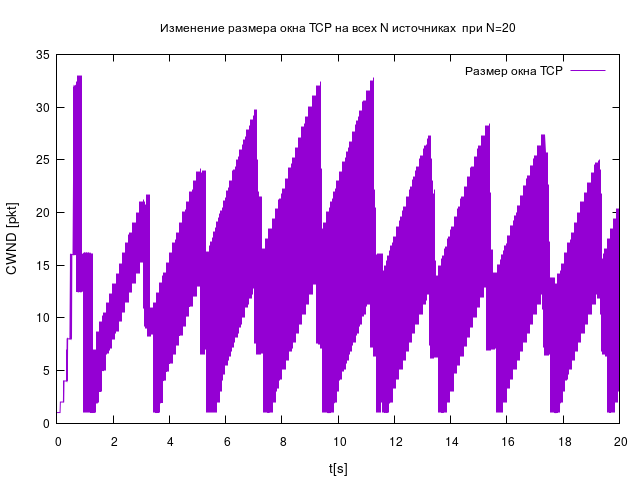


Рис. 7: Изменение размера окна TCP на всех источниках при N=20

Графики изменения размера длины очереди (рис. 8) и размера средней длины очереди (рис. 9).

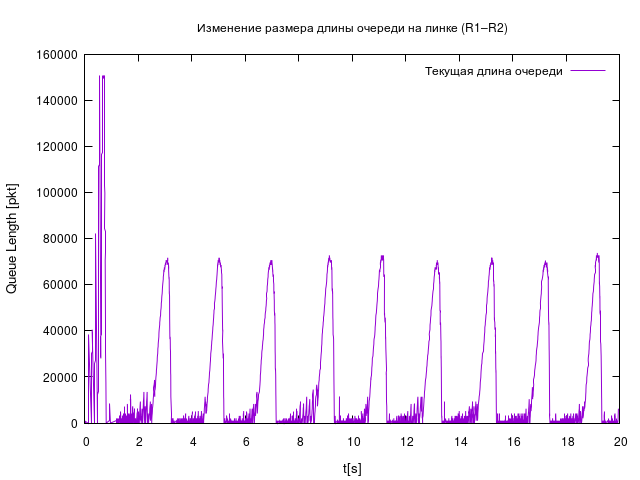


Рис. 8: Изменение размера длины очереди на линке (R1–R2) при N=20

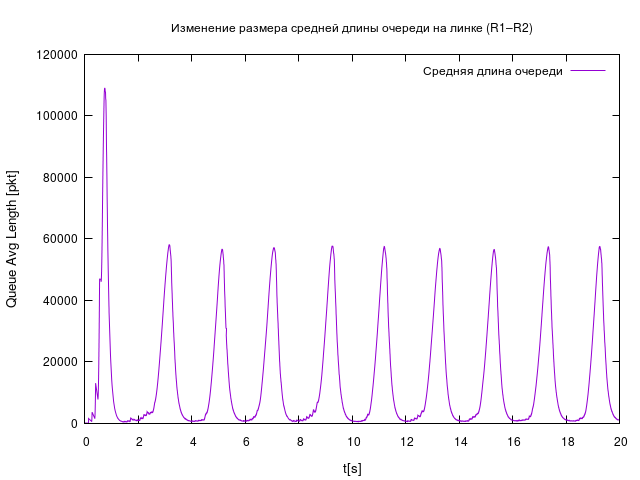


Рис. 9: Изменение размера средней длины очереди на линке (R1–R2) при N=20

# 4 Выводы

В результате выполнения данной лабораторной работы была разработана имитационная модель в пакете NS-2, построены графики изменения размера окна TCP, изменения длины очереди и средней длины очереди.