

Отчёт по лабораторной работе 4

Задания для самостоятельного выполнения

Сидорова Наталья Андреевна

Содержание

1	Выполнение лабораторной работы	5
2	Выводы	17
	Список литературы	18

Список иллюстраций

1.1	Создание модели	6
1.2	Код моделирования	7
1.3	Моделирование сети	8
1.4	Сама модель	8
1.5	График на одном источнике	9
1.6	График на всех источниках	10
1.7	График очереди	11
1.8	График средней очереди	12
1.9	Код в GNUPlot	13
1.10	Код в GNUPlot 2	14
1.11	График средней очереди	14
1.12	График очереди	15
1.13	График на одном источнике	15
1.14	График на всех источниках	16

Список таблиц

1 Выполнение лабораторной работы

По заданию создала модель сети, состоящей из 20 ТСП источников и 20 ТСП приемников, двух маршрутизаторов между приемниками и источниками. Между ТСП источниками и первым маршрутизатором установлены дуплексные соединения с пропускной способностью 100 Мбит/с, задержкой 20мс и очередью DropTail. Между ТСП приемниками и вторым маршрутизатором установлены дуплексные соединения с пропускной способностью 100 Мбит/с, задержкой 20мс и очередью DropTail. Между маршрутизаторами установлено симплексное соединение с пропускной способностью 20 Мбит/с, задержкой 15 мс и очередью типа RED, размер буфера 300 пакетов. В обратную сторону установлено симплексное соединение с пропускной способностью 15 Мбит/с, задержкой 20 мс и очередью типа DropTail (рис. 1.1).

```

# 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: Red"
    close $f

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

    exec awk $awkCode all.q

```

Рис. 1.1: Создание модели

(рис. 1.2).

```

# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -fg pink -bg green -bb -tk -x time -t "TCPReNoCWND" WindowVsTimeRenoOne &
exec xgraph -fg pink -bg green -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
}

# 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 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]
}

# 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;

```

Рис. 1.2: Код моделирования

(рис. 1.3).

```

# 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

```

Рис. 1.3: Моделирование сети

Получилась модель, подходящая требованиям (рис. 1.4).

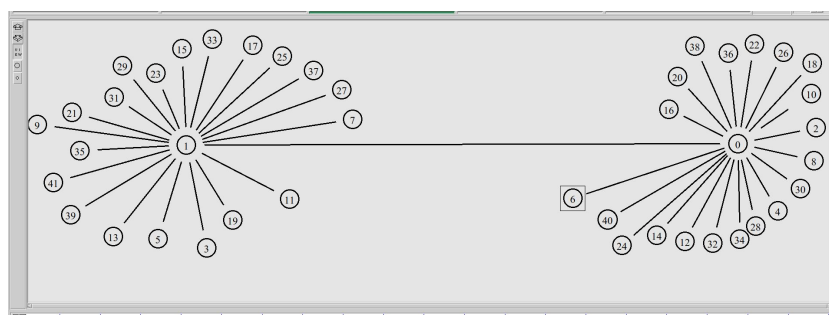


Рис. 1.4: Сама модель

График Xgraph изменения размера TCP окна на линке одного источника (рис. 1.5).

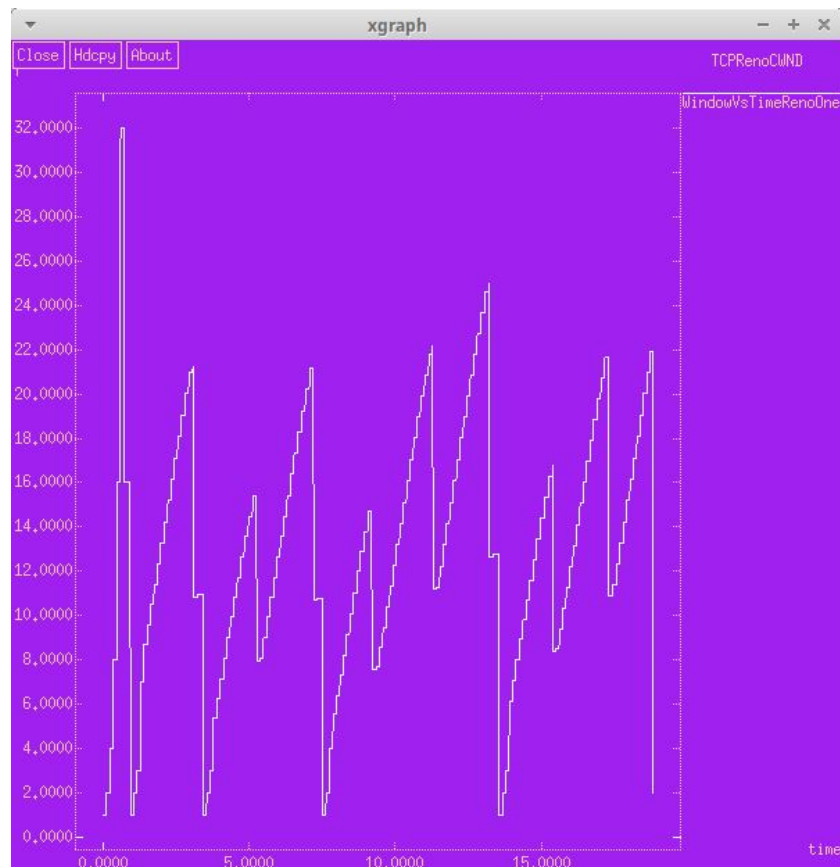


Рис. 1.5: График на одном источнике

График Xgraph изменения размера TCP окна на всех источниках (рис. 1.6).

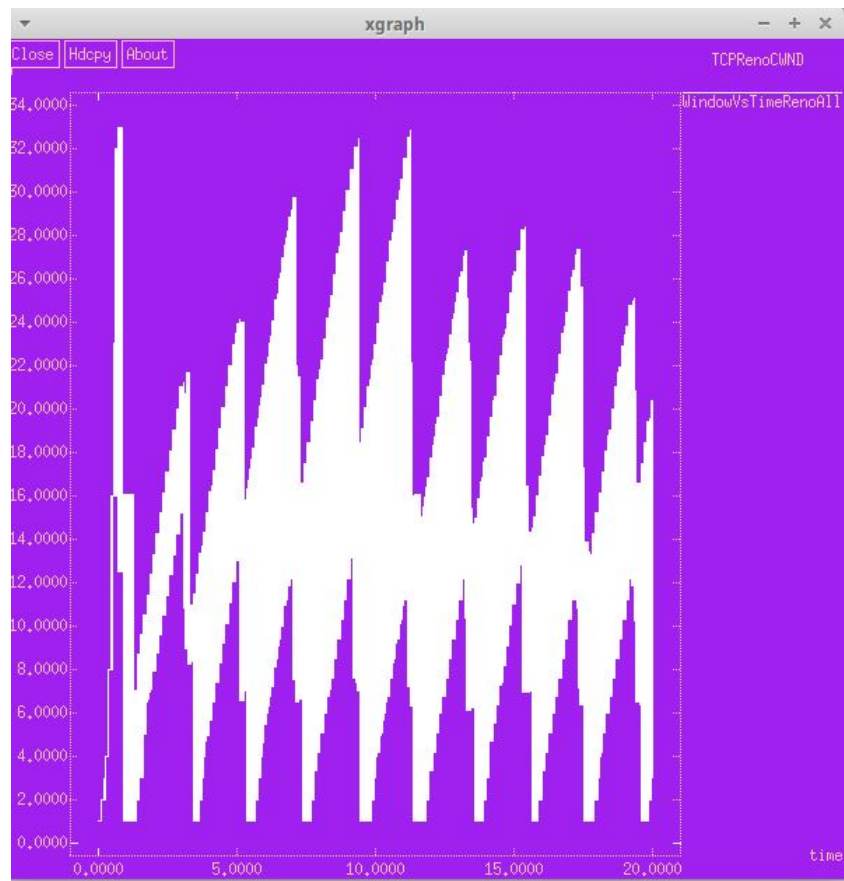


Рис. 1.6: График на всех источниках

График Xgraph изменения размера очереди (рис. 1.7).

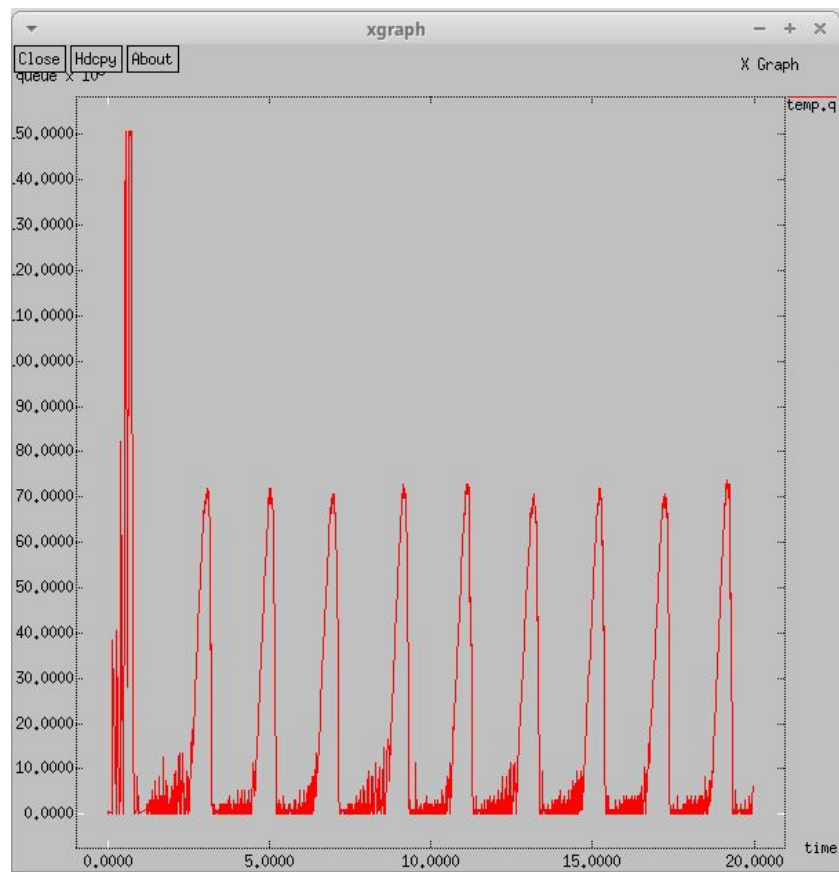


Рис. 1.7: График очереди

График Xgraph изменения размера средней очереди (рис. 1.8).

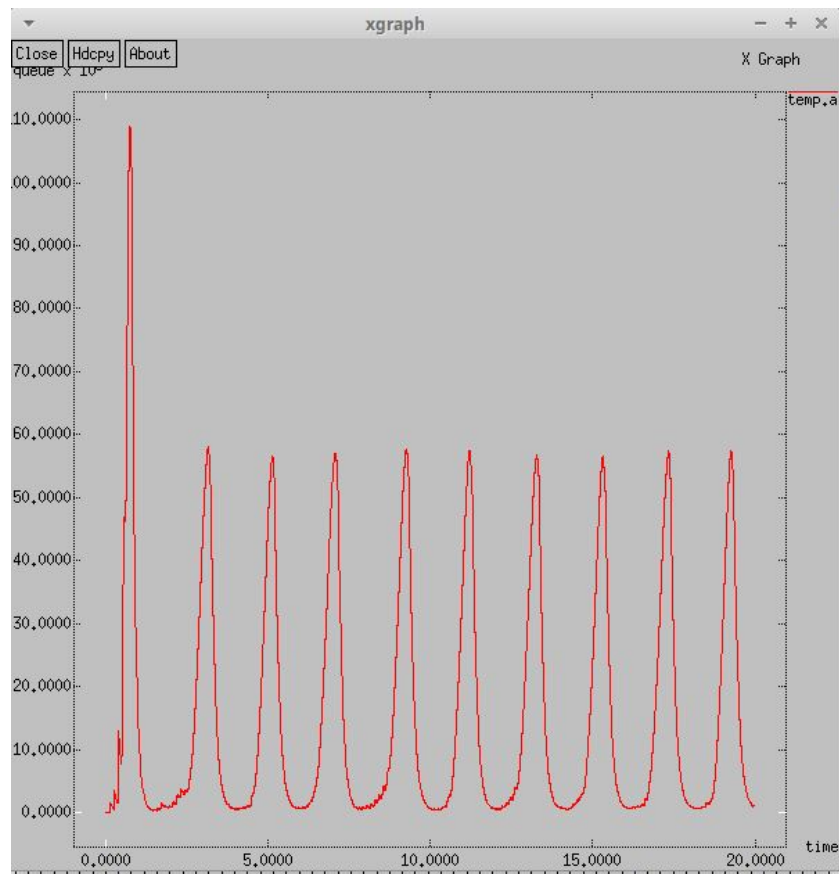


Рис. 1.8: График средней очереди

Создала те же графики, но в GNUPlot (рис. 1.9).

```

#!/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)"

```

Рис. 1.9: Код в GNUPlot

(рис. 1.10).

```

# 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. 1.10: Код в GNUPlot 2

График GNUPlot изменения размера средней очереди (рис. 1.11).

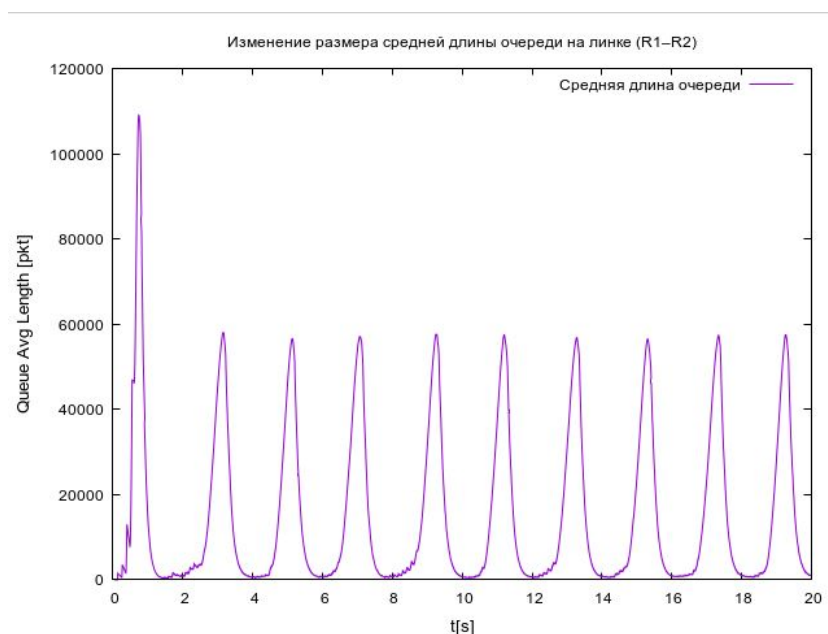


Рис. 1.11: График средней очереди

График GNUPlot изменения размера очереди (рис. 1.12).

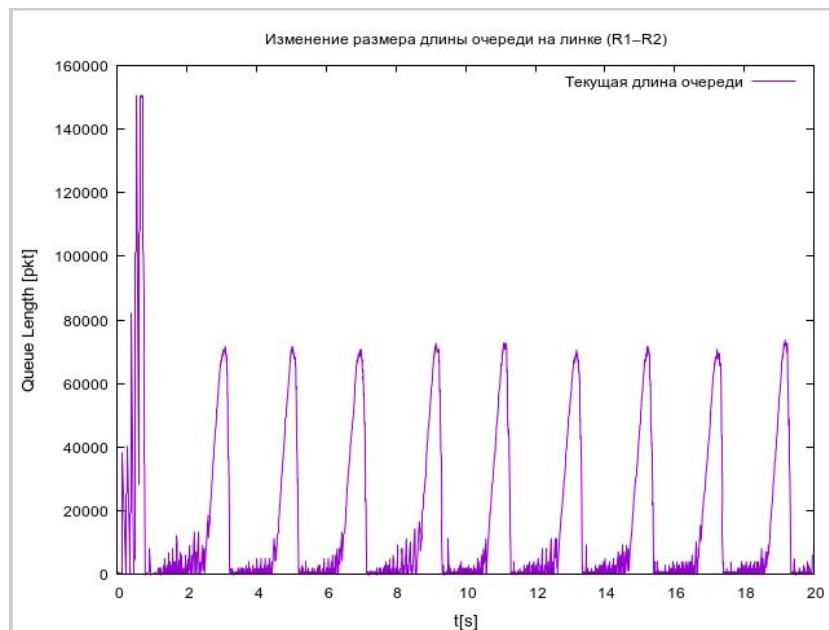


Рис. 1.12: График очереди

График GNUPlot изменения размера TCP окна на линке одного источника (рис. 1.13).

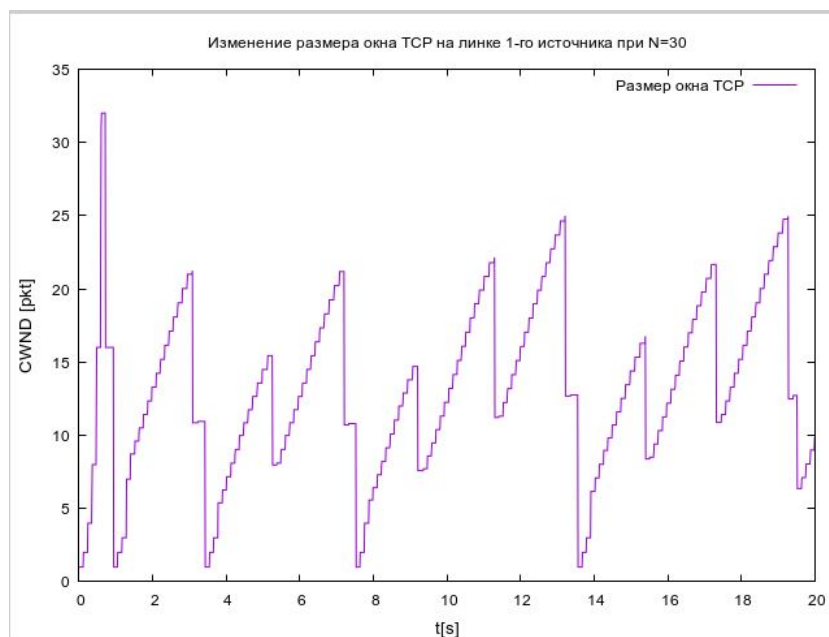


Рис. 1.13: График на одном источнике

График GNUPlot изменения размера TCP окна на всех источниках (рис. 1.14).

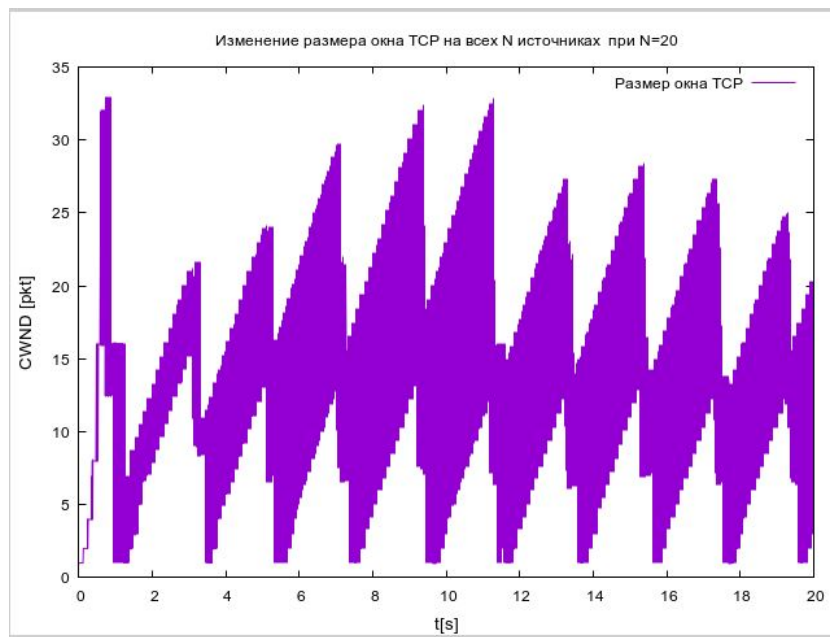


Рис. 1.14: График на всех источниках

2 Выводы

Создала свою сеть в NS-2, смоделировала передачу данных по моей сети, создала графики изменения длины очереди, средней длины очереди, изменения размера TCP окна на линке одного источника, изменения размера TCP окна на всех источниках в GNUPlot и Xgraph

Список литературы