

# **Лабораторная работа 4**

**Задание для самостоятельного выполнения**

Ланцова Яна Игоревна

# **Содержание**

<b>1 Цель работы</b>	<b>4</b>
<b>2 Задание</b>	<b>5</b>
<b>3 Описание моделируемой сети</b>	<b>6</b>
<b>4 Выполнение лабораторной работы</b>	<b>7</b>
<b>5 Выводы</b>	<b>15</b>

# Список иллюстраций

4.1	код программы . . . . .	8
4.2	код программы . . . . .	8
4.3	код программы . . . . .	9
4.4	схема сети . . . . .	9
4.5	Изменение размера длины очереди на линке ( $R_1-R_2$ ) при $N=40$ . . . . .	10
4.6	Изменение размера окна TCP на всех источниках при $N=40$ . . . . .	11
4.7	Изменение размера средней длины очереди на линке ( $R_1-R_2$ ) при $N=40$ . . . . .	11
4.8	код программы . . . . .	12
4.9	код программы . . . . .	12
4.10	Изменение размера средней длины очереди на линке ( $R_1-R_2$ ) при $N=40$ . . . . .	13
4.11	Изменение размера длины очереди на линке ( $R_1-R_2$ ) при $N=40$ . . . . .	13
4.12	Изменение размера окна TCP на линке 1-го источника при $N=40$ . . . . .	14
4.13	Изменение размера окна TCP на всех источниках при $N=40$ . . . . .	14

# **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 GNUMPlot);
3. Построить график изменения длины очереди и средней длины очереди на первом маршрутизаторе;
4. Оформить отчёт о выполненной работе.

### **3 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:  $q_{\min} = 75$ ,  $q_{\max} = 150$ ,  $q_w = 0,002$ ,  $p_{\max} = 0.1$ ;
- максимальный размер TCP-окна 32; размер передаваемого пакета 500 байт; время моделирования – не менее 20 единиц модельного времени.

## 4 Выполнение лабораторной работы

Откроем файл .tcl на редактирование, в нем построим сеть. Зададим  $N = 40$  TCP-источников,  $N = 40$  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:  $q_{min} = 75$ ,  $q_{max} = 150$ ,  $q_w = 0,002$ ,  $p_{max} = 0.1$ . Также нам нужно выполнить мониторинг окна TCP и мониторинг очереди. Программа выглядит следующим образом: (рис. [fig:001?] - [fig:003?]).

```

* файл Правка Поиск Вид Документ Справка
# создано объектом Simulator
set ns [new Simulator]
# открытие на запись файла out.nam для анимализатора пакетов
set nf [open out.nam w]
# все результаты моделирования будут записаны в переменную nf
$ns namtrace-all $nf
# открытие на запись файла гравировок 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
    # параллельное 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.a w]
    puts $f "0.Color: Purple"
    close $f
    set f [open temp.a w]
    puts $f "0.Color: Purple"
    close $f
    exec awk $awkCode all.q
}

```

Рис. 4.1: код программы

```

# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -fg pink -bg purple -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeRenoOne &
exec xgraph -fg pink -bg purple -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 &
exec rm -f temp.q 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"
}

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 40
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. 4.2: код программы

```

# Мониторинг очереди:
$redq set [$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 currq_
$redq trace ave_
$redq attach $tchan_

for {set i 0} {$i < $N} {incr i} {
    $ns at 0.0 "$!lp($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. 4.3: код программы

Сеть имеет вид (рис. [fig:004?]).

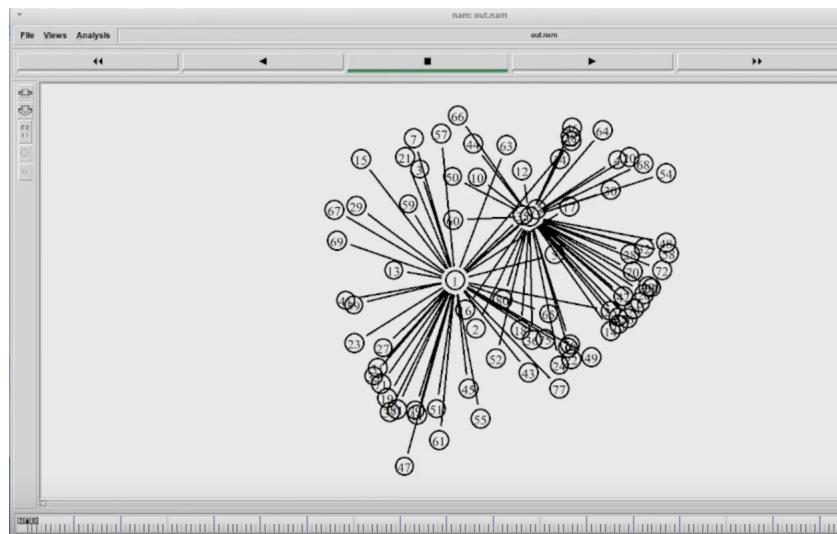


Рис. 4.4: схема сети

В результате получим следующие графики (рис. [fig:005?] - [fig:007?]).

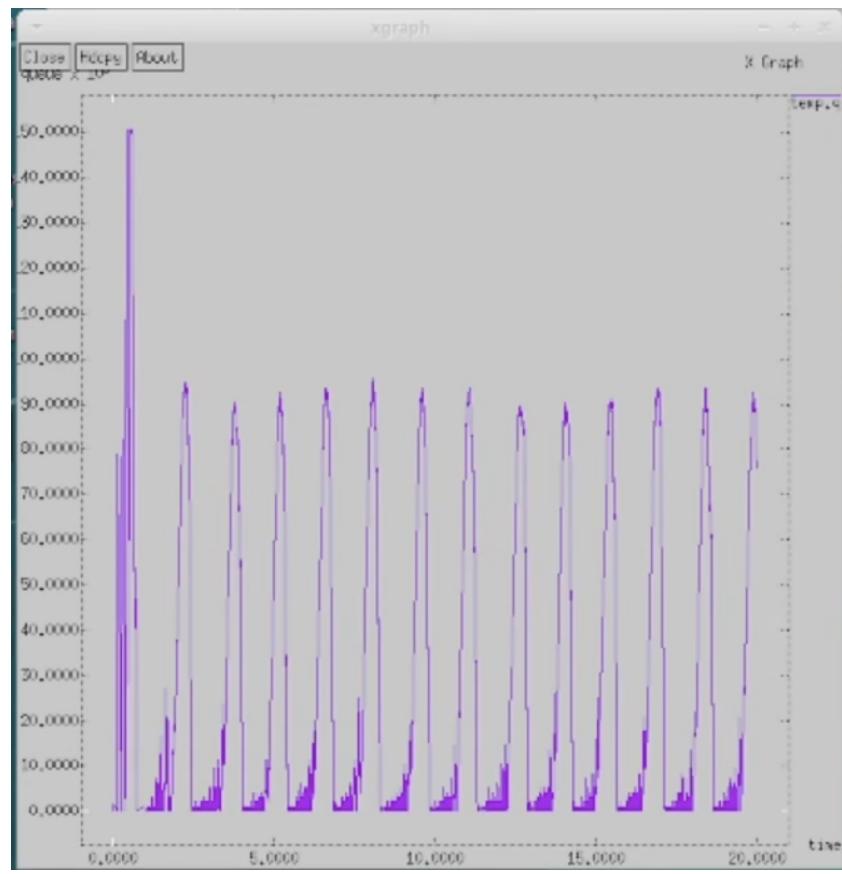


Рис. 4.5: Изменение размера длины очереди на линке (R1 – R2) при N=40

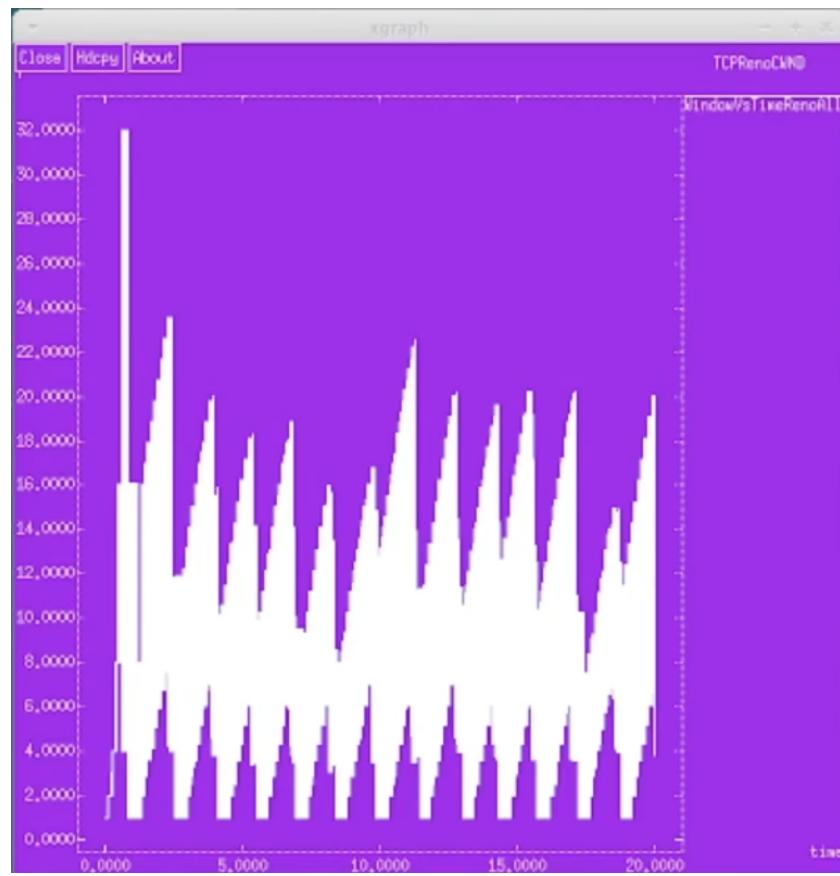


Рис. 4.6: Изменение размера окна TCP на всех источниках при N=40

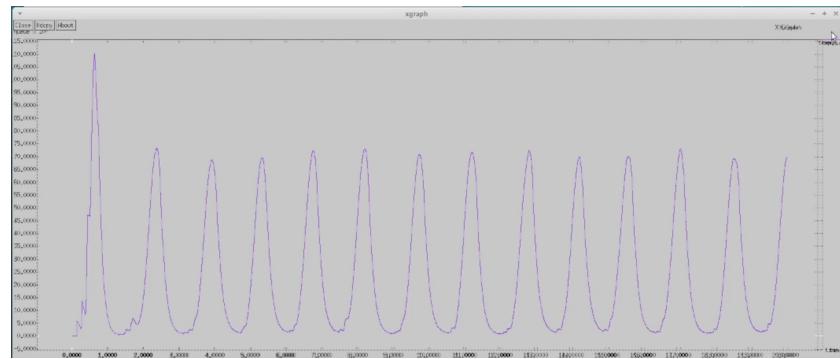


Рис. 4.7: Изменение размера средней длины очереди на линке (R1–R2) при N=40

Напишем программу для построения графиков в GNUPlot: (рис. [fig:008?] - [fig:009?]).

```

/home/openmodelica/mip/lab-ns/graph_plot_lab04 - Mousepad
Файл Правка Поиск Вид Документ Справка
#!/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 'result01.png'

# задаём название графика
set title "Изменение размера окна TCP на линке 1-го источника"

# 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 'result02.png'

# задаём название графика
set title "Изменение размера окна TCP на всех N источниках"

# 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)"

```

Рис. 4.8: код программы

```

# 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)"

# 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. 4.9: код программы

Сделаем исполняемым и запустим его. Получим 4 графика. (рис. [fig:010?] -

[fig:013?]).

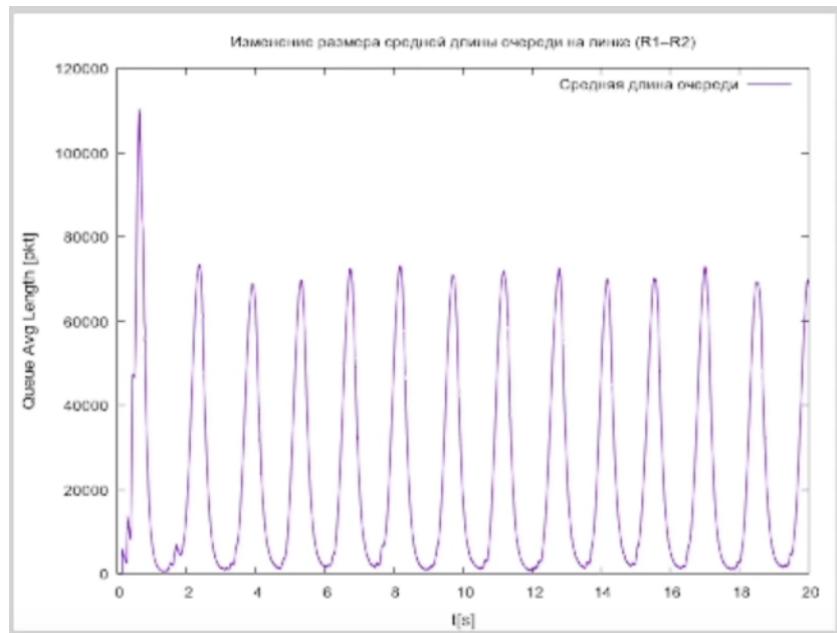


Рис. 4.10: Изменение размера средней длины очереди на линке (R1–R2) при  $N=40$

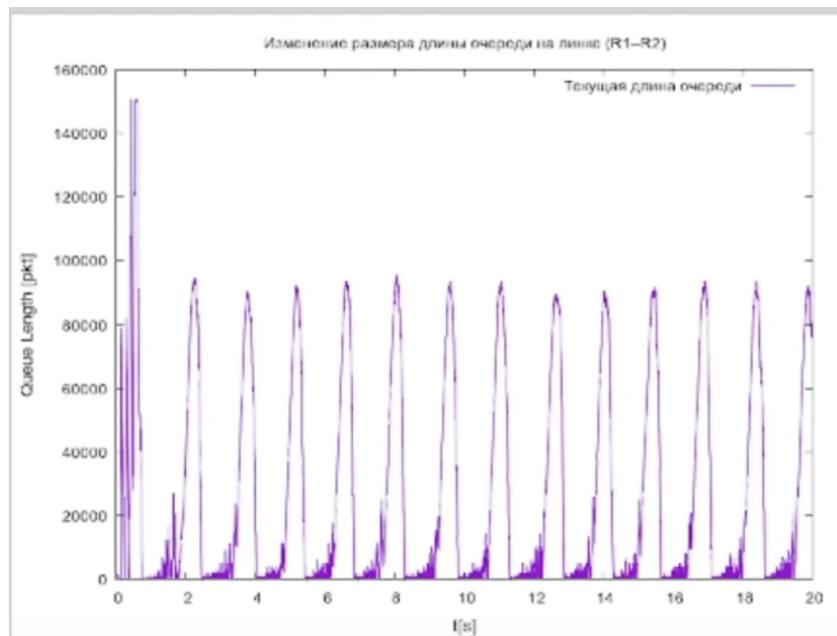


Рис. 4.11: Изменение размера длины очереди на линке (R1–R2) при  $N=40$

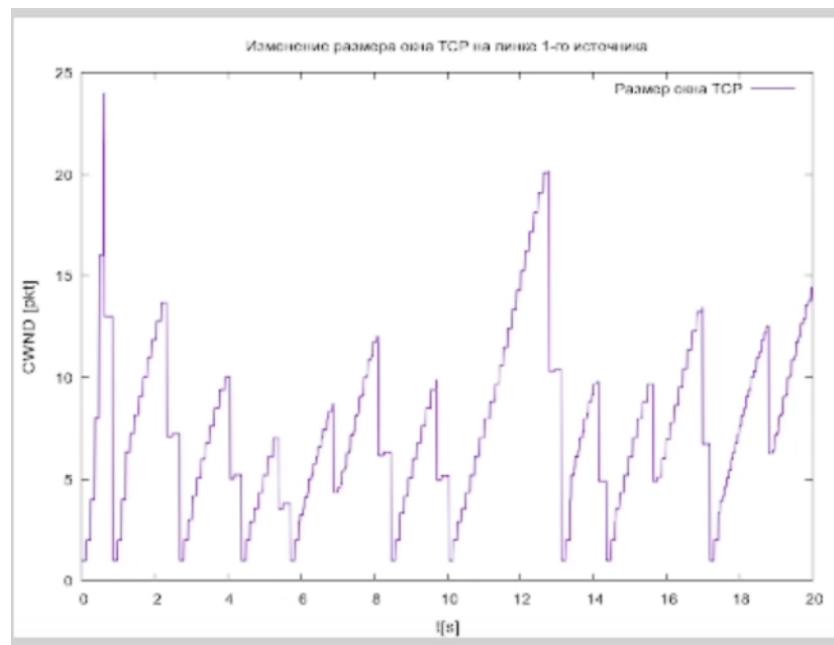


Рис. 4.12: Изменение размера окна TCP на линке 1-го источника при N=40

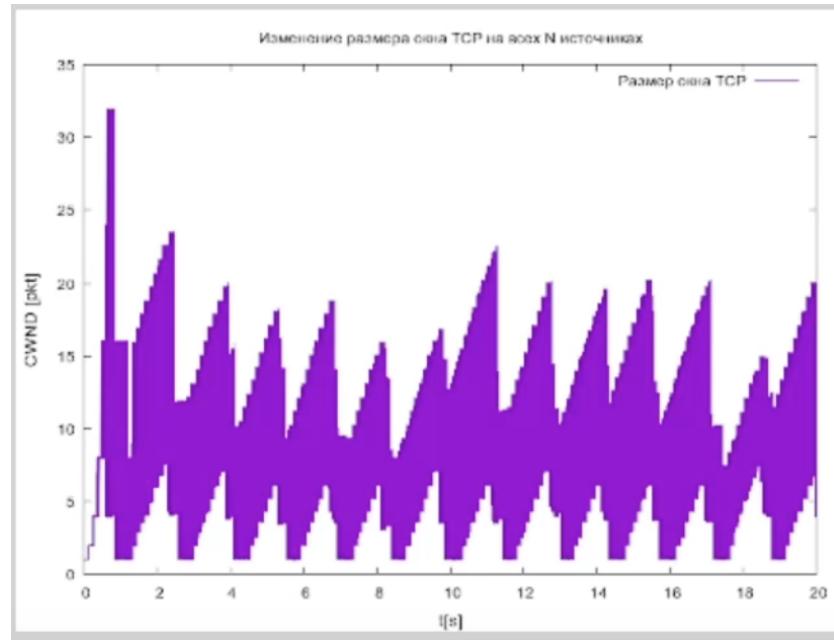


Рис. 4.13: Изменение размера окна TCP на всех источниках при N=40

## **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 GNUMplot).