

# **Лабораторная работа №4**

**Задание для самостоятельного выполнения**

Астраханцева А. А.

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>

## Список иллюстраций

3.1	Модель описываемой сети . . . . .	11
3.2	Графики изменения размера окна . . . . .	12
3.3	Графики изменения размера длины очереди . . . . .	12
3.4	Графики изменения размера окна в GNUplot . . . . .	15
3.5	Графики изменения размера длины очереди в GNUplot . . . . .	15

# 1 Цель работы

Закрепление навыков моделирования сетей с помощью средства имитационного моделирования NS-2, а также построения графиков с помощью GNUplot.

## 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 TCP Reno;
- 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.

### Описание написанного скрипта

Для начала создам директорию для файлов четвертой ЛР. В ней создала файл script.tcl, в котором написала скрипт для моделирования сети, описанной выше.

Параметр N и время моделирования взяла за 25. В данном скрипте сначала создается объект Simulator, необходимые файлы для записи результатов. Описываются процедуры finish и plotWindow, запускается xgraph для отрисовки графиков изменения размера окна TCP на линке 1-го источника, изменения размера окна TCP на всех источниках, изменения размера длины очереди на линке (R1–R2) и изменения среднего размера длины очереди на линке (R1–R2). Также происходит создание самой цепи: 2 маршрутизатора с разными типами очереди и TCP-приемники, TCP-источники с параметрами, заданными в задании. В данном скрипте ведется мониторинг размера длины очереди и размера окна.

```
# 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]
#close $f

#set f [open temp.a w]
#close $f

exec awk $awkCode all.q

# Запуск xgraph с графиками окна TCP и очереди:
# Изменение размера окна на одном источнике
exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowSizeOne &
# Изменение размера окна на всех источниках

```



```

exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowSizeAll &
# Изменение длины очереди на первом маршрутизаторе источниках
exec xgraph -bb -tk -x time -y queue temp.q &
# Изменение средней длины очереди на первом маршрутизаторе источниках
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 25
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 windowSizeOne [open WindowSizeOne w]
set windowSizeAll [open WindowSizeAll w]

set qmon [$ns monitor-queue $r1 $r2 [open qm.out w] 0.1];
[$ns link $r1 $r2] queue-sample-timeout;

# 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) $windowSizeAll"
}

$ns at 0.0 "plotWindow $tcp(1) $windowSizeOne"

# at-событие для планировщика событий, которое запускает
$ns at 25.0 "finish"

# запуск модели
$ns run

```

После написания этого скрипт можно запустить командой `ns script.tcl` и мы увидим 4 появившихся графика и окно аниматора `nam` (рис. 3.1).

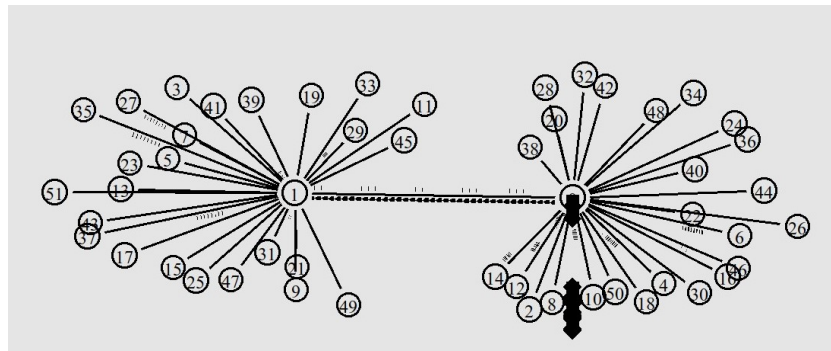


Рис. 3.1: Модель описываемой сети

Графики изменения размера окна TCP на линке 1-го источника, изменения размера окна TCP на всех источниках (рис. 3.2).

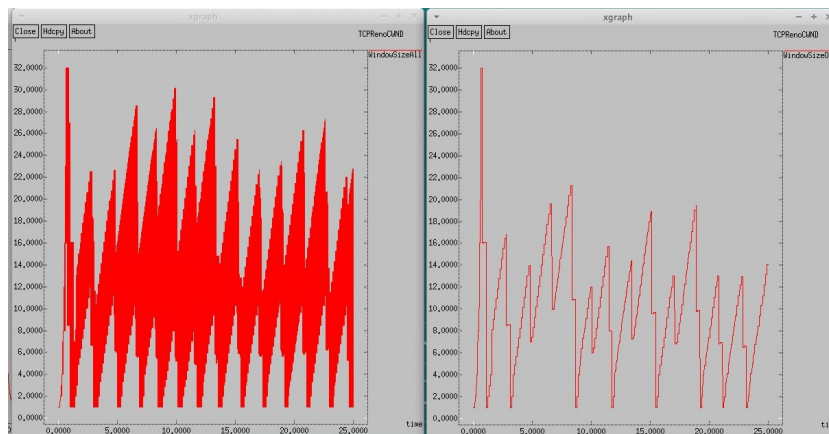


Рис. 3.2: Графики изменения размера окна

Графики изменения размера длины очереди на линке (R1–R2) и изменения среднего размера длины очереди на линке (R1–R2) (рис. 3.2).

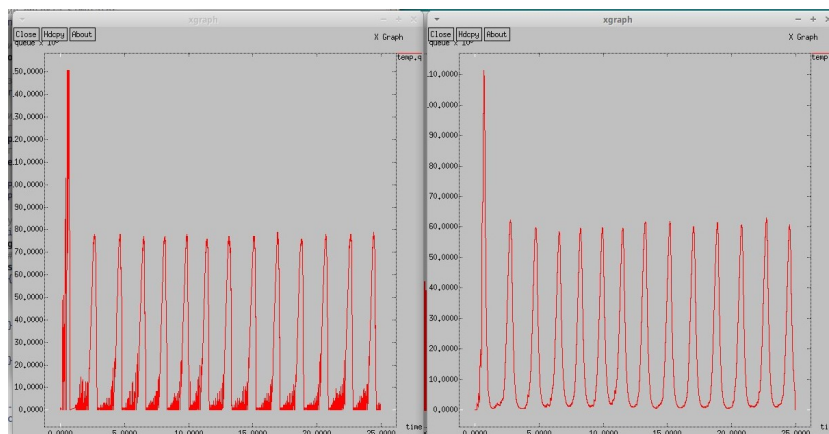


Рис. 3.3: Графики изменения размера длины очереди

Далее нужно было построить эти же графики, но уже с помощью GNUpot, для этого мной был написан такой скрипт:

```
#!/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_one.pdf'

# задаём название графика
set title "Изменение размера окна TCP на линке 1-го источника при N=25"

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

# задаём выходной файл графика
set out 'window_all.pdf'

# задаём название графика
set title "Изменение размера окна TCP на всех источниках при N=25"

# построение графика, используя значения
# 1-го и 2-го столбцов файла WindowSizeAll
plot "WindowSizeAll" 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)"

# 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.pdf'

# 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"

```

Этот скрипт я сохранила в файле `gnuplot_l4`. Для запуска нужно было добавить право на исполнение и запустить скрипт:

```

chmod +x gnuplot_l4
./gnuplot_l4

```

В ходе того, как отработала данная программа были созданы 4 графика: графики изменения размера окна TCP на линке 1-го источника, изменения размера окна TCP на всех источниках (рис. 3.4).

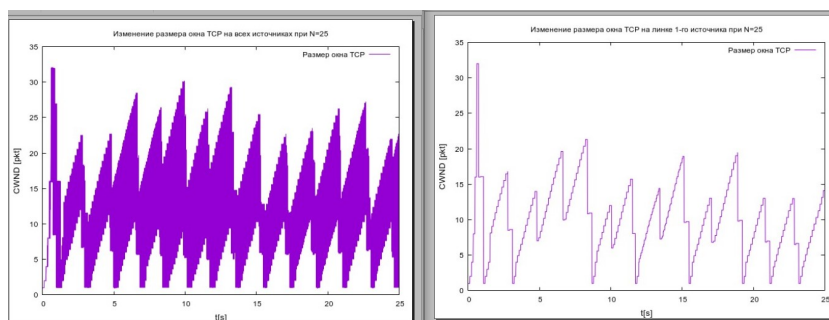


Рис. 3.4: Графики изменения размера окна в GNUplot

Графики изменения размера длины очереди на линке (R1–R2) и изменения среднего размера длины очереди на линке (R1–R2) (рис. 3.5).

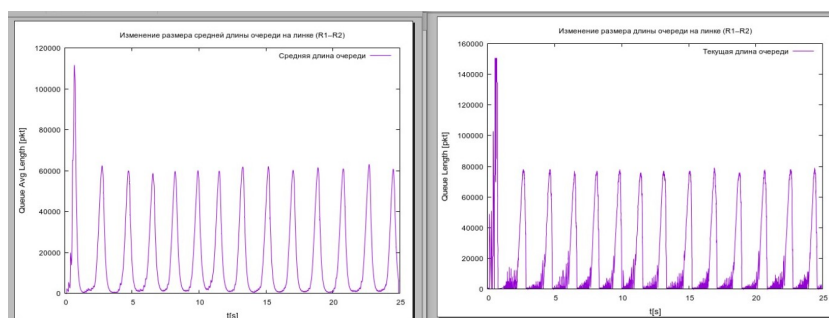


Рис. 3.5: Графики изменения размера длины очереди в GNUplot

## 4 Выводы

В ходе выполнения лабораторной работы я закрепила навыки моделирования сетей с помощью средства имитационного моделирования NS-2, а также построения графиков с помощью GNUplot.