

Отчёт по лабораторной работе №4

Задание для самостоятельного выполнения

Козлов Всеволод Павлович НФИбд-02-22

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	18

Список иллюстраций

3.1	пат файл моделируемой сети	11
3.2	График изменения окна TCP на линке 1-го источника при $N = 30$.	12
3.3	График изменения окна TCP на всех источниках при $N = 30$	12
3.4	График изменения размера длины очереди при $N = 30$	13
3.5	График изменения средней длины очереди при $N = 30$	13
3.6	График изменения размера окна TCP на линке 1-го источника при $N = 30$	16
3.7	График изменения размера окна TCP на всех источниках при $N = 30$	16
3.8	График изменения размера длины очереди при $N = 30$	17
3.9	график изменения размера средней длины очереди при $N = 30$. .	17

Список таблиц

1 Цель работы

Выполнить самостоятельно задание при помощи NS-2, GNUplot, Xgraph.

2 Задание

1. Для приведенной схемы разработать имитационную модель в пакеты NS-2.
2. Построить график изменения размера окна TCP.
3. Построить график изменения длины очереди и средней длины очереди на первом маршрутизаторе.
4. Оформить отчёт о выполненной работе.

3 Выполнение лабораторной работы

Создал файл lab4_2.tcl. И записал в него код, приведенный ниже:

```
# 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: Purple"
close $f

```

```

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

```

```

exec awk $awkCode all.q

```

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 &
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 30
```

```
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;

```

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

```

Запустил программу с помощью NS-2. Получил nam файл со схемой моделируемой сети (рис. 3.1)

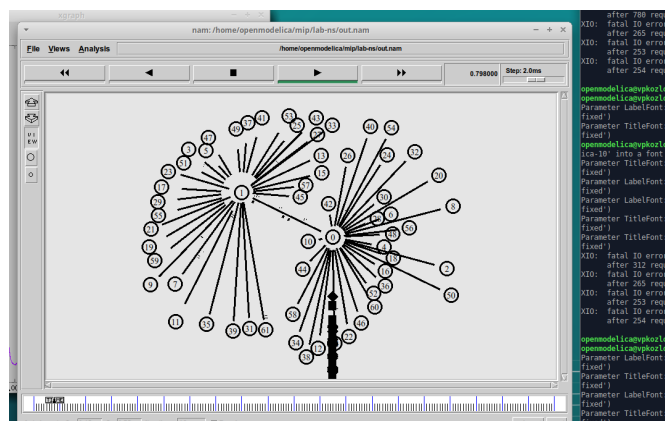


Рис. 3.1: nam файл моделируемой сети

Получил график изменения окна TCP на линке 1-го источника с помощью Xgraph (рис. 3.2)

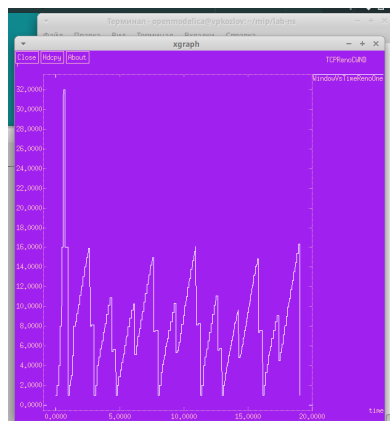


Рис. 3.2: График изменения окна TCP на линке 1-го источника при $N = 30$

Получил график изменения окна TCP на всех источниках с помощью Xgraph (рис. 3.3)

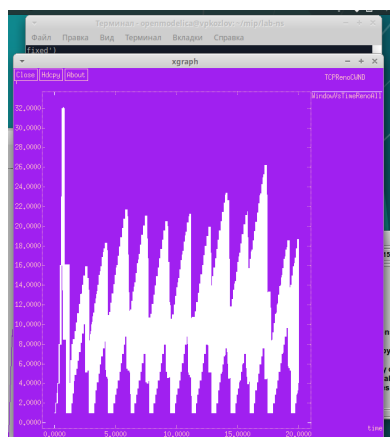


Рис. 3.3: График изменения окна TCP на всех источниках при $N = 30$

Получил график изменения размера длины очереди с помощью Xgraph (рис. 3.4)

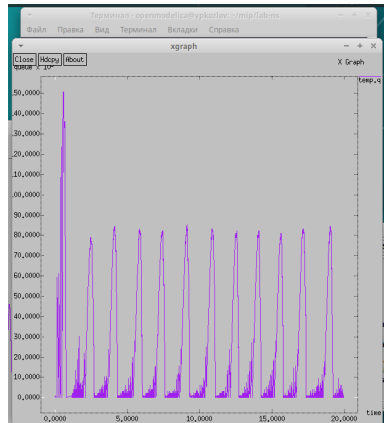


Рис. 3.4: График изменения размера длины очереди при $N = 30$

Получил график изменения средней длины очереди с помощью Xgraph (рис. 3.5)

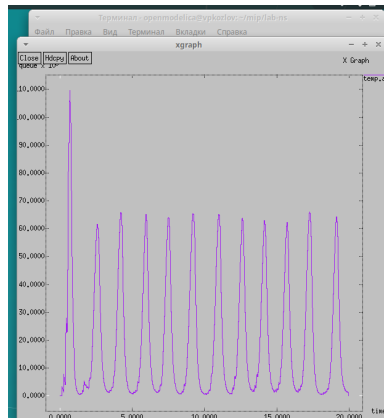


Рис. 3.5: График изменения средней длины очереди при $N = 30$

Создал файл lab4_2 и записал в него код для Gnuplot-графиков, представленный ниже. Сделал его исполняемым и запустил

```
#!/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)"
```

```
# 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"
```

Получил график изменения размера окна TCP на линке 1-го источника (рис. 3.6)

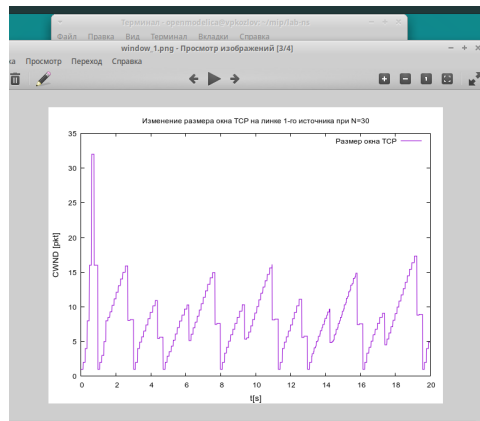


Рис. 3.6: График изменения размера окна TCP на линке 1-го источника при $N = 30$

Получил график изменения размера окна TCP на всех источниках (рис. 3.7)

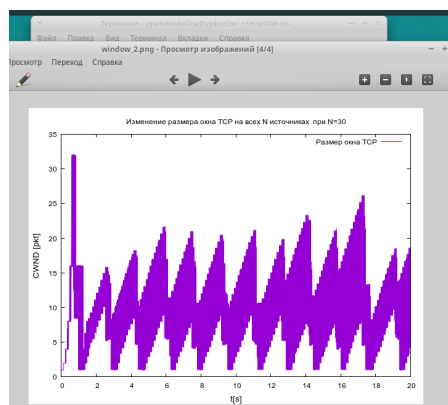


Рис. 3.7: График изменения размера окна TCP на всех источниках при $N = 30$

Получил график изменения размера длины очереди (рис. 3.8)

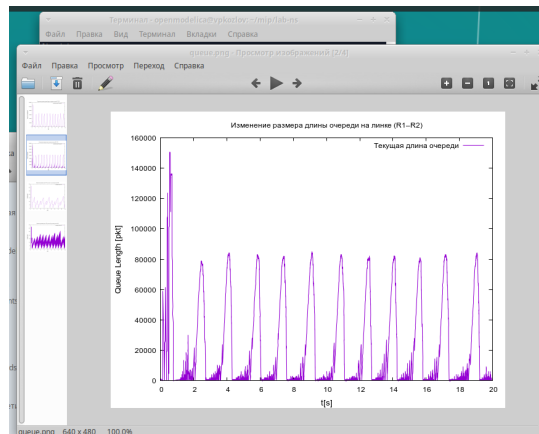


Рис. 3.8: График изменения размера длины очереди при $N = 30$

Получил график изменения размера средней длины очереди (рис. 3.9)

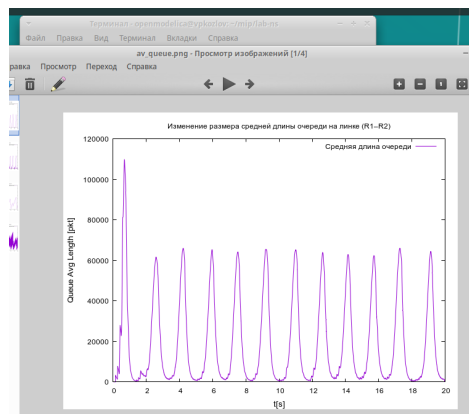


Рис. 3.9: график изменения размера средней длины очереди при $N = 30$

4 Выводы

Выполнил самостоятельное задание. Применил NS-2, GnuPlot, Xgraph.