

Задание для самостоятельного выполнения

Извекова Мария ПЕТровна

Содержание

Список иллюстраций

Список таблиц

Цель работы

Выполнить задание для самостоятельного выполнения.

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.

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

Создаем файл .tcl для построения сети. Зададим $N = 22$ TCP-источников, $N = 2$ TCP-приёмников, два маршрутизатора r1 и r2 между источниками и приёмниками. Между TCP-источниками и первым маршрутизатором устанавливаем дуплексные соединения с пропускной способностью 100 Мбит/с и задержкой 20 мс очередью типа DropTail; между TCP-приёмниками и вторым маршрутизатором устанавливаем 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: $q_{min} = 75$, $q_{max} = 150$, $q_w = 0,002$, $p_{max} = 0.1$. Данный скрипт будет выглядеть следующим образом

```
1 # создание объекта Simulator
2 set ns [new Simulator]
3
4 # открытие на запись файла out.nam для визуализатора nam
5 set nf [open out.nam w]
6
7 # все результаты моделирования будут записаны в переменную nf
8 $ns namtrace-all $nf
9
10 # открытие на запись файла трассировки out.tr
11 set f [open out.tr w]
12 $ns trace-all $f
13
14 # максимальный размер TCP-окна 32; размер передаваемого пакета 500 байт
15 Agent/TCP set window_ 32
16 Agent/TCP set pktSize_ 500
17
18 # Процедура finish:
19 proc finish {} {
20     global tchan_
21     # подключение кода AWK:
22     set awkCode {
23     {
24     if ($1 == "Q" && NF>2) {
25     print $2, $3 >> "temp.q";
26     set end $2
27     }
28     else if ($1 == "a" && NF>2)
29     print $2, $3 >> "temp.a";
30     }
31 }
```

```

#создание файла куда будет записываться средняя длина очереди и изменение очереди
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 # выполнение кода AWK

# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -fg pink -bg purple -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeNewRenoOne &
exec xgraph -fg pink -bg purple -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeNewRenoAll &
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
}

```

```

56 # Формирование файла с данными о размере окна TCP:
57 proc plotWindow {tcpSource file} {
58     global ns
59     set time 0.01
60     set now [$ns now]
61     set cwnd [$tcpSource set cwnd_]
62     puts $file "$now $cwnd"
63     $ns at [expr $now+$time] "plotWindow $tcpSource $file"
64 }
65
66 set r1 [$ns node]
67 set r2 [$ns node]
68
69 $ns simplex-link $r1 $r2 20Mb 15ms RED
70 $ns simplex-link $r2 $r1 15Mb 20ms DropTail
71 $ns queue-limit $r1 $r2 300
72
73
74 # Узлы сети:
75 set N 22
76 for {set i 0} {$i < $N} {incr i} {
77     set n1($i) [$ns node]
78     $ns duplex-link $n1($i) $r1 100Mb 20ms DropTail
79     set n2($i) [$ns node]
80     $ns duplex-link $n2($i) $r2 100Mb 20ms DropTail
81
82     set tcp($i) [$ns create-connection TCP/Reno $n1($i) TCPSink $n2($i) $i]
83     set ftp($i) [$tcp($i) attach-source FTP]
84 }

```

```

85
86 # Мониторинг размера окна TCP:
87 set windowVsTimeOne [open WindowVsTimeRenoOne w]
88 puts $windowVsTimeOne "0.Color: White"
89 set windowVsTimeAll [open WindowVsTimeRenoAll w]
90 puts $windowVsTimeAll "0.Color: White"
91
92 set qmon [$ns monitor-queue $r1 $r2 [open qm.out w] 0.1];
93 [$ns link $r1 $r2] queue-sample-timeout;
94
95 # Мониторинг очереди:
96 set redq [$ns link $r1 $r2] queue
97 $redq set thresh_ 75
98 $redq set maxthresh_ 150
99 $redq set q_weight_ 0.002
100 $redq set linterm_ 10

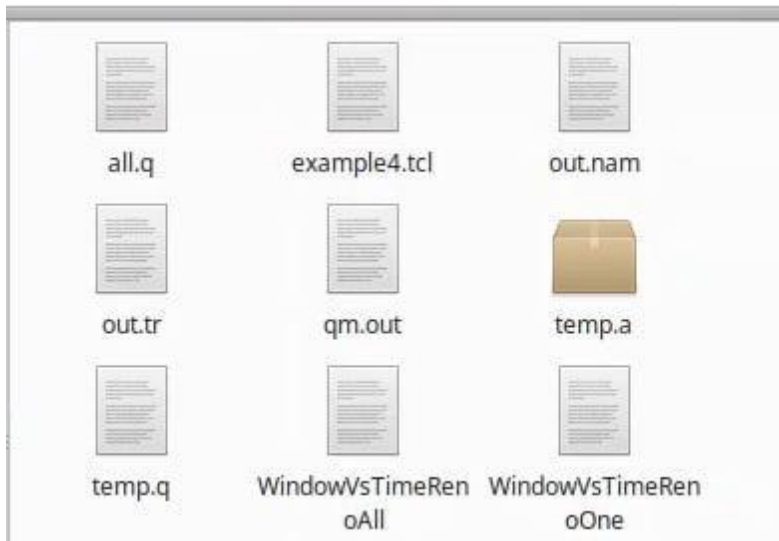
```

После запуска файла .tcl получаем файлы с нашей симуляцией и графиками

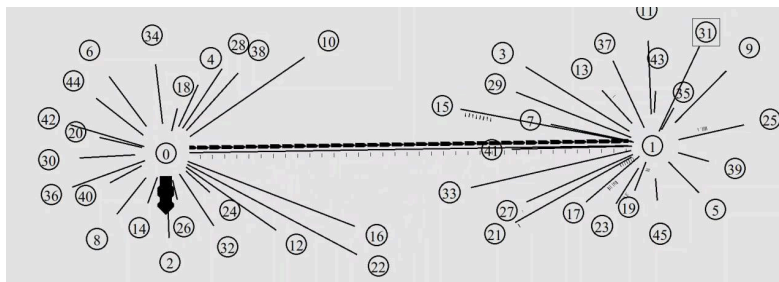
```

openmodelica@openmodelica-VirtualBox:~/mip/lab-ns/lab4$ ns example4.tcl
ns: finish: awk: командная строка:7: критическая ошибка: невозможно открыть файл
`#' для чтения (Нет такого файла или каталога)
while executing
"exec awk $awkCode all.q # выполнение кода AWK"
(procedure "finish" line 27)
invoked from within
"finish"

```

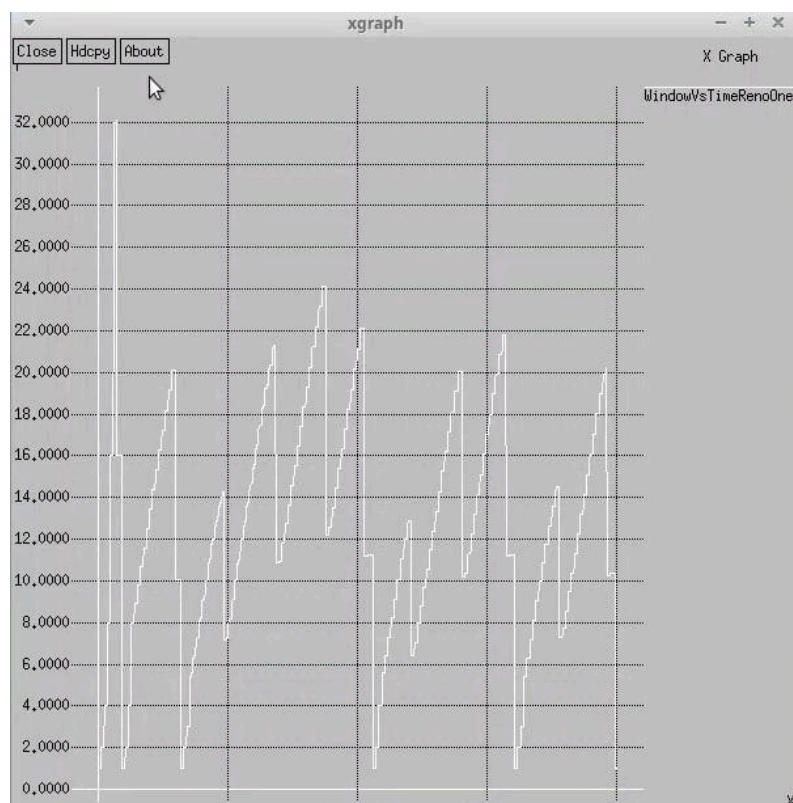


Запускаем файл .nam симуляцию и получаем схему моделируемой сети

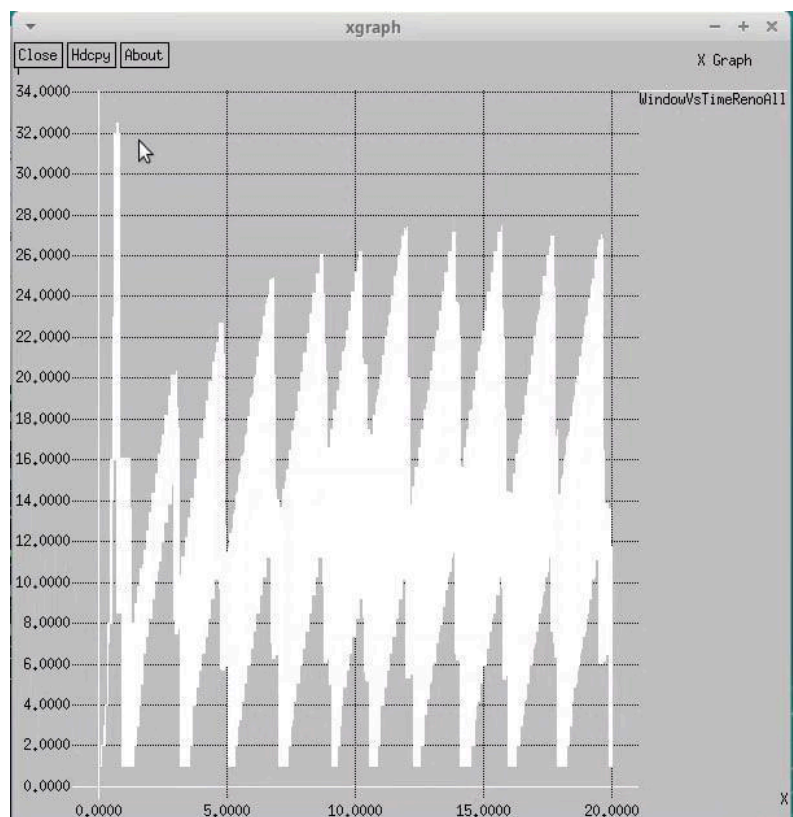


Полученный графики:

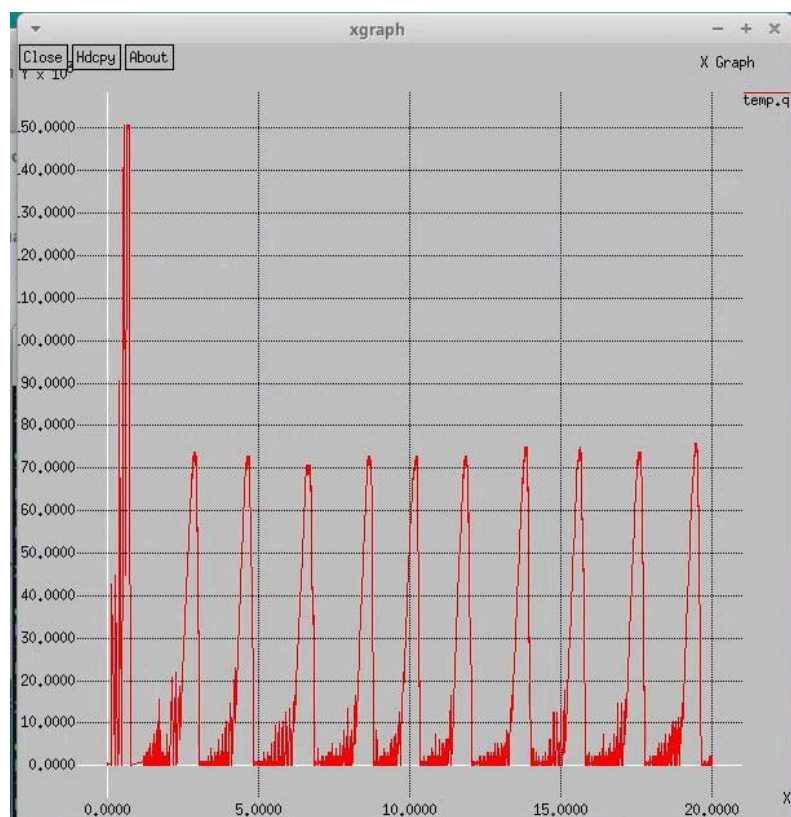
Изменение размера окна TCP на линке 1-го источника при N=22



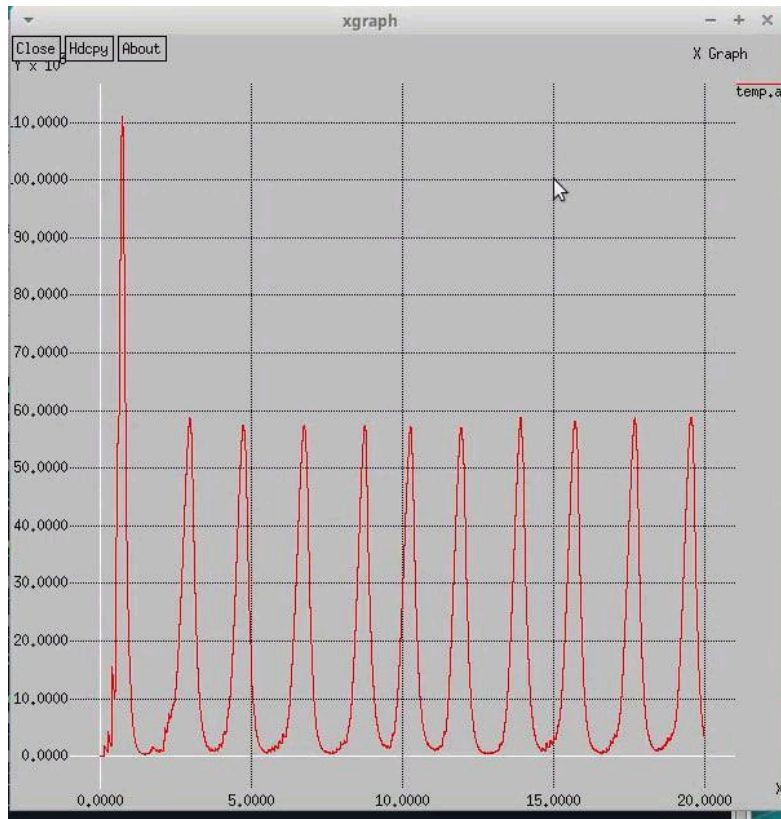
Изменение размера окна TCP на всех источниках при N=22



Изменение размера длины очереди на линке (R1–R2) при $N=20$, $q_{\min} = 75$, $q_{\max} = 150$



Изменение размера средней длины очереди на линке (R1–R2) при $N=20$, $q_{\min} = 75$, $q_{\max} = 150$



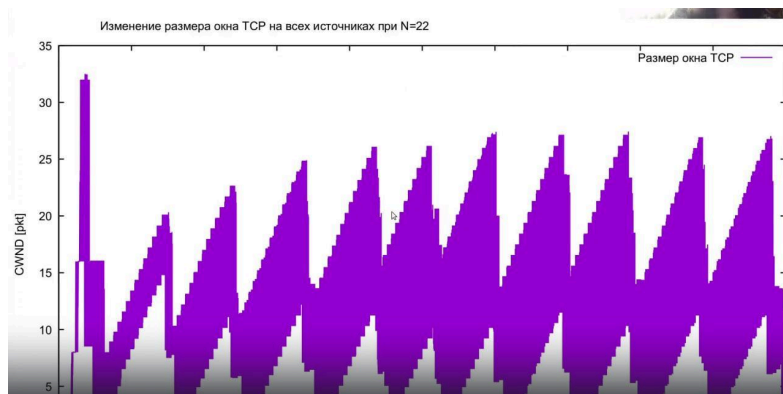
Далее напишем скрипт для построение графиков в GNUPlot

```

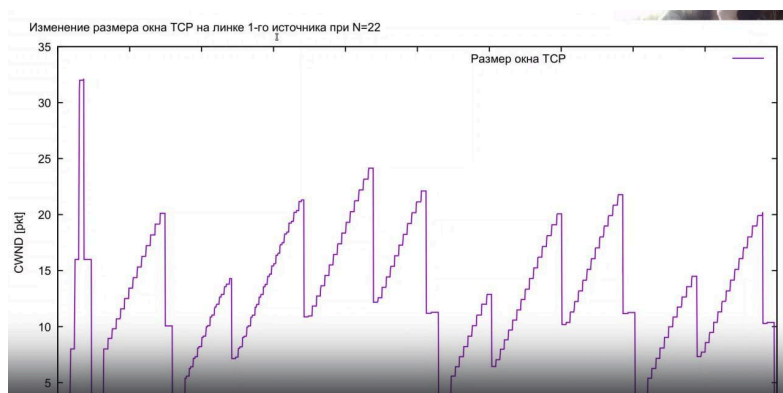
Файл  Правка  Поиск  Вид  Документ  Справка
1 #!/usr/bin/gnuplot -persist
2 set encoding utf8
3 set term pdfcairo font "Arial,9"
4
5 set out 'qm.pdf'
6 set title "Изменение размера окна TCP на линке 1-го источника при N=22"
7 set xlabel "t [s]" font "Arial,9"
8 set ylabel "cwnd [pkt]" font "Arial,9"
9 plot "WindowVsTimeRenoOne" using ($1):($2) with lines title "Размер окна TCP"
10
11 set out 'qm2.pdf'
12 set title "Изменение размера окна TCP на всех источниках при N=22"
13 plot "WindowVsTimeRenoAll" using ($1):($2) with lines title "Размер окна TCP"
14
15 set out 'qm.png'
16 set title "4. Изменение размера длины очереди на линке (R1-R2)"
17 set xlabel "t [s]" font "Arial,9"
18 set ylabel "Queue length [pkt]" font "Arial,9"
19 plot "temp.q" using ($1):($2) with lines title "Текущая длина очереди"
20
21 set out 'qm2.png'
22 set title "Изменение размера средней длины очереди на линке (R1-R2)"
23 set xlabel "t [s]" font "Arial,9"
24 set ylabel "Queue avg length [pkt]" font "Arial,9"
25 plot "temp.a" using ($1):($2) with lines title "Средняя длина очереди"

```

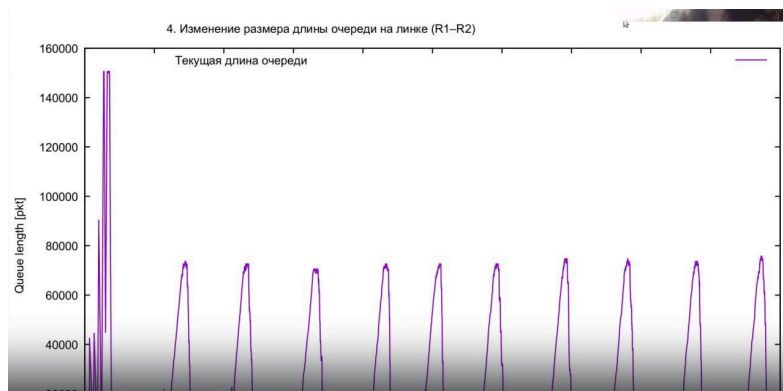
Результаты:



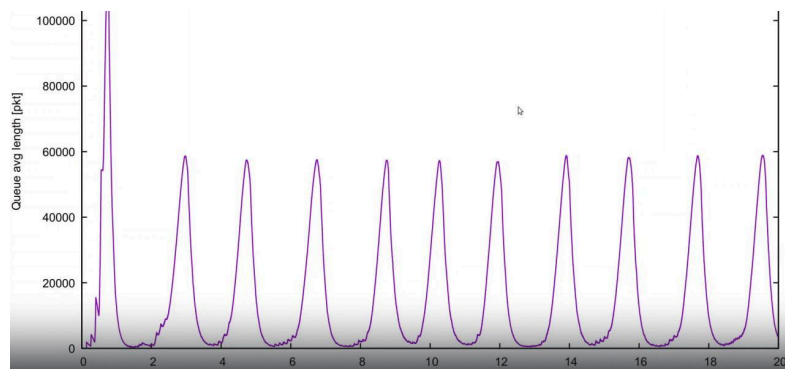
Изменение размера окна TCP на всех источниках при N=22



Изменение размера окна TCP на линке 1-го источника при N=22



Изменение размера длины очереди на линке



Изменение размера средней длины очереди на линке

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.