Отчет по лабораторной работе №2

Дисциплина: Имитационное моделирование

Лобанова Полина Иннекентьевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Исследование протокола TCP и алгоритма управления очередью RED.

2 Задание

1. Описание моделируемой сети:

– сеть состоит из 6 узлов;

– между всеми узлами установлено дуплексное соединение с различными пропускной способностью и задержкой 10 мс;

– узел r1 использует очередь с дисциплиной RED для накопления пакетов, максимальный размер которой составляет 25;

– TCP-источники на узлах s1 и s2 подключаются к TCP-приёмнику на узле s3;

– генераторы трафика FTP прикреплены к TCP-агентам.

2. – Измените в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas. Сравните и поясните результаты.

– Внесите изменения при отображении окон с графиками (измените цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде).

3 Выполнение лабораторной работы

1. Скопировала шаблон в новый файл и дополнила его.

```
openmodelica@openmodelica-VirtualBox:~$ cd mip/lab-ns
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ls
ex1.tcl  example1.tcl  example2.tcl  example3.tcl  out.nam  out.tr  shablon.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ touch lab2_1.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ls
ex1.tcl       example2.tcl  lab2_1.tcl  out.tr
example1.tcl  example3.tcl  out.nam     shablon.tcl
```

Рис. 3.1: *Копирование файла*

```
set ns [new Simulator]

# Узлы сети:
set N 5
for {set i 1} {$i < $N} {incr i} {
        set node_(s$i) [$ns node]
}
set node_(r1) [$ns node]
set node_(r2) [$ns node]

# Соединения:
$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail

# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Reno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeReno w]
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[$ns link $node_(r1) $node_(r2)] queue-sample-timeout;

# Мониторинг очереди:
set redq [[$ns link $node_(r1) $node_(r2)] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_

# Добавление at-событий:
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"

proc plotWindow {tcpSource file} {
        global ns
        set time 0.01
        set now [$ns now]
        set cwnd [$tcpSource set cwnd_]
```

Рис. 3.2: *Заполнение файла*

```
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"
}

# 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";
                }
        }

        set f [open temp.queue w]
        puts $f "TitleText: red"
        puts $f "Device: Postscript"

        if { [info exists tchan_] } {
                close $tchan_
        }

        exec rm -f temp.q temp.a
        exec touch temp.a temp.q
        exec awk $awkCode all.q
        puts $f \"queue
        exec cat temp.q >@ $f
        puts $f \n\"ave_queue
        exec cat temp.a >@ $f
        close $f

        # Запуск xgraph с графиками окна TCP и очереди:
        exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno &
        exec xgraph -bb -tk -x time -y queue temp.queue &
        exit 0
}

$ns run
```

Рис. 3.3: *Заполнение файла*

Рис. 3.4: *График изменения длины очереди и средней длины очереди*

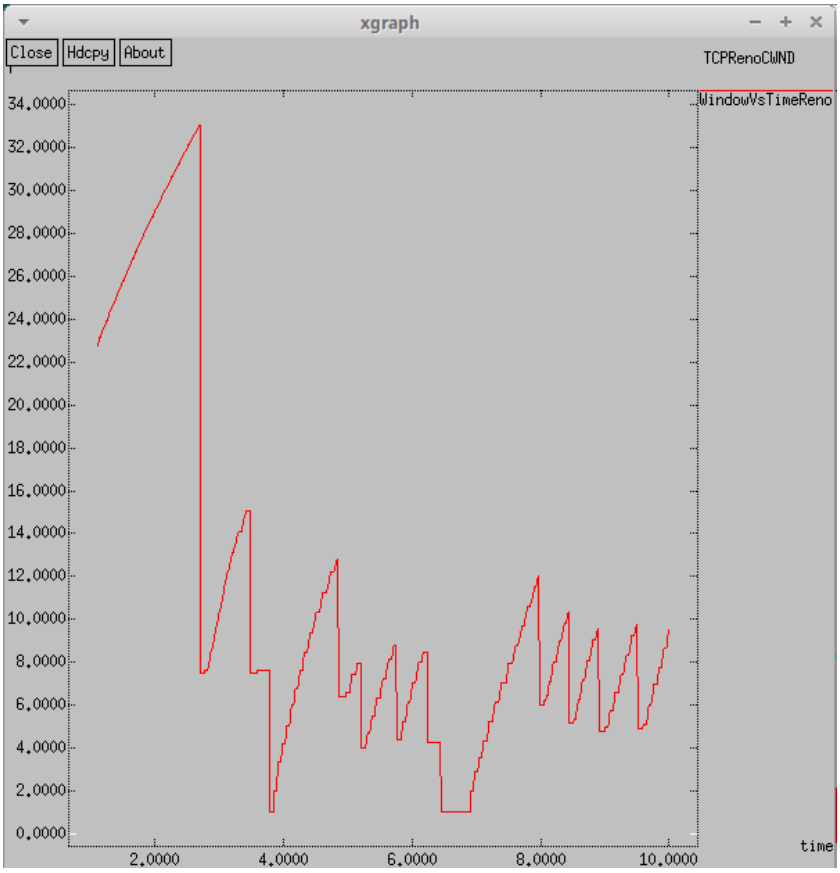

Рис. 3.5: *График изменения TCP-окна*

2. Изменила в модели на узле s1 тип протокола TCP с Reno на NewReno.

```
# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Newreno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
```

Рис. 3.6: *Изменение типа протокола*

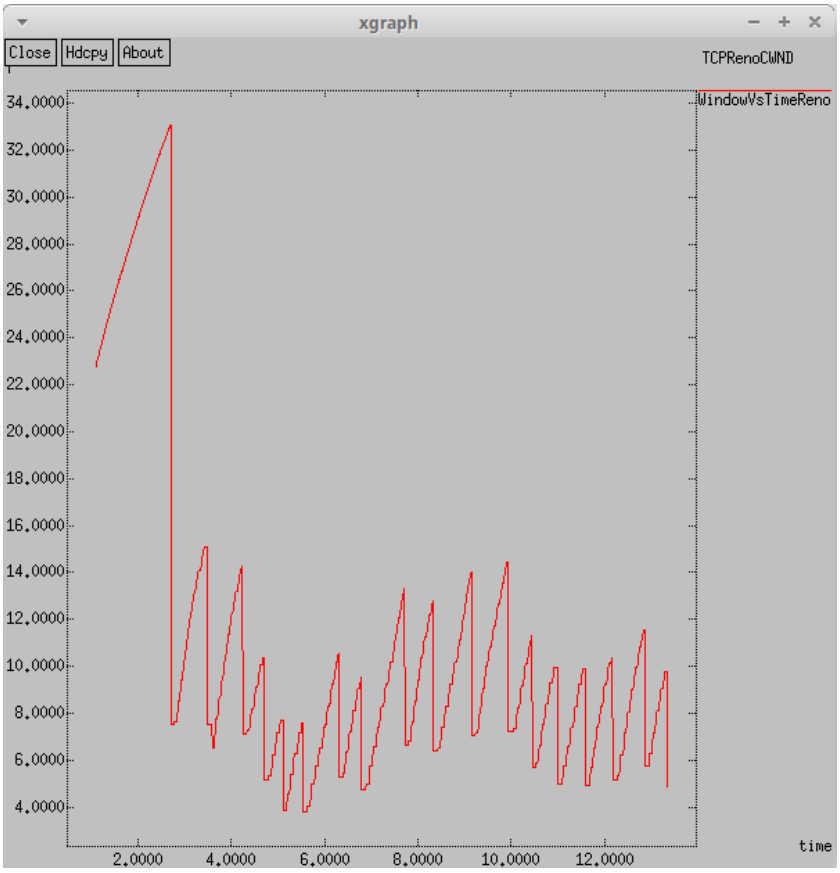

Рис. 3.7: *График изменения TCP-окна*

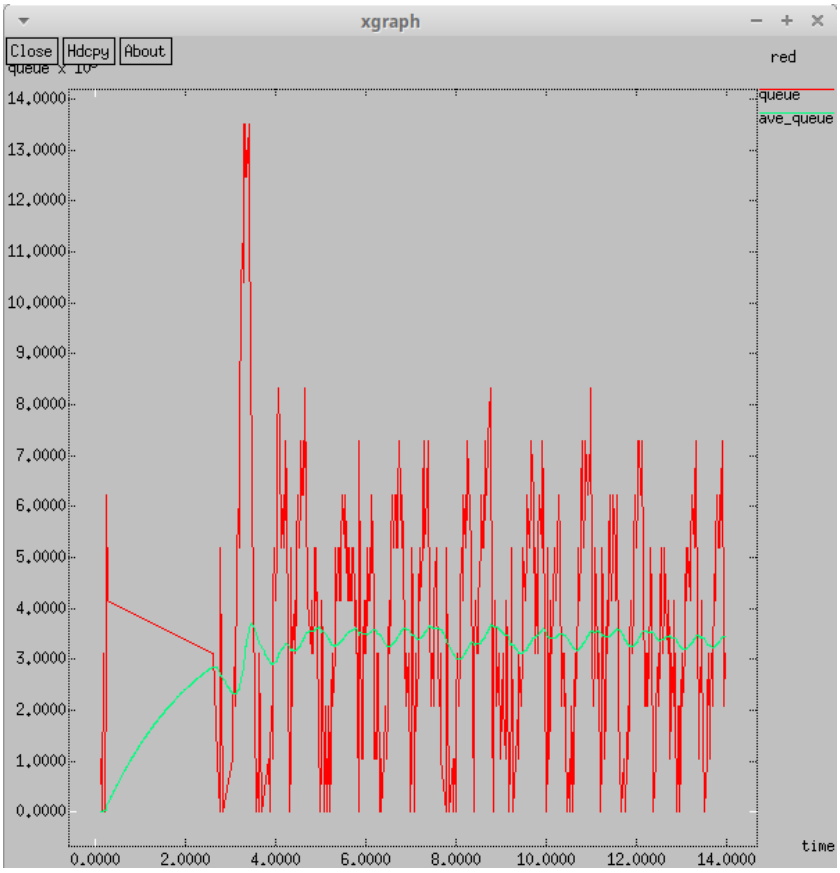

Рис. 3.8: *График изменения длины очереди и средней длины очереди*

3. Изменила в модели на узле s1 тип протокола TCP с NewReno на Vegas.

```
# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Vegas $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
```

Рис. 3.9: *Изменение типа протокола*

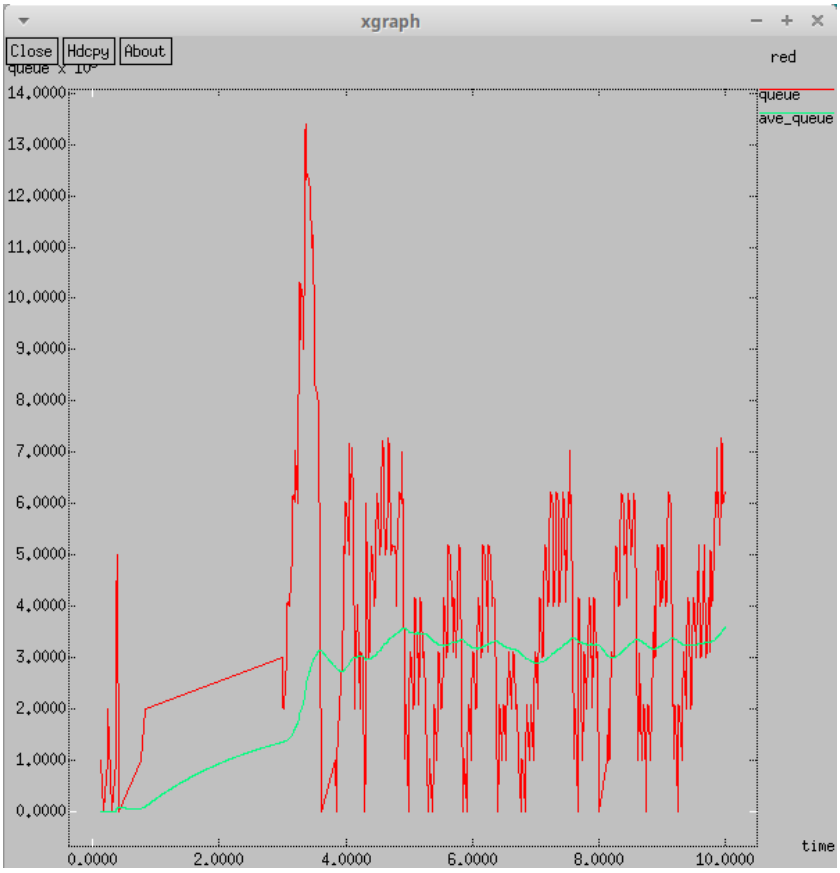

Рис. 3.10: *График изменения длины очереди и средней длины очереди*

4. Внесла изменения при отображении окон с графиками (изменила цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде).

```
set f [open temp.queue w]
puts $f "TitleText: red"
puts $f "Device: Postscript"
puts $f "0.Color: White"
puts $f "0.Color: Grey"
if { [info exists tchan_] } {
        close $tchan_
}

exec rm -f temp.q temp.a
exec touch temp.a temp.q
exec awk $awkCode all.q
puts $f \"QUEUE
exec cat temp.q >@ $f
puts $f \n\"AVE_QUEUE
exec cat temp.a >@ $f
close $f
# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -fg green -bg pink -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno &
exec xgraph -fg purple -bg pink -bb -tk -x TIME -y QUEUE temp.queue &
exit 0
```

Рис. 3.11: *Изменение графика*

Рис. 3.12: *График изменения длины очереди и средней длины очереди*

4 Выводы

Я исследовала протокол TCP и алгоритм управления очередью RED.