

Лабораторная работа №2

Дисциплина: Имитационное моделирование

Пронякова Ольга Максимовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11
	Список литературы	12

Список иллюстраций

3.1	Результат выполнения программы с Reno	7
3.2	изменение в коде	8
3.3	Изменение в коде	9
3.4	Результат выполнения программы с NewReno	10
3.5	Результат выполнения программы с Vegas	10

Список таблиц

1 Цель работы

Исследовать протокол TCP и алгоритм управления очередью RED.

2 Задание

- 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

Требуется разработать сценарий, реализующий модель согласно рис. 2.4, построить в Xgraph график изменения TCP-окна, график изменения длины очереди и средней длины очереди. Используя код из инструкции(рис.3.1).

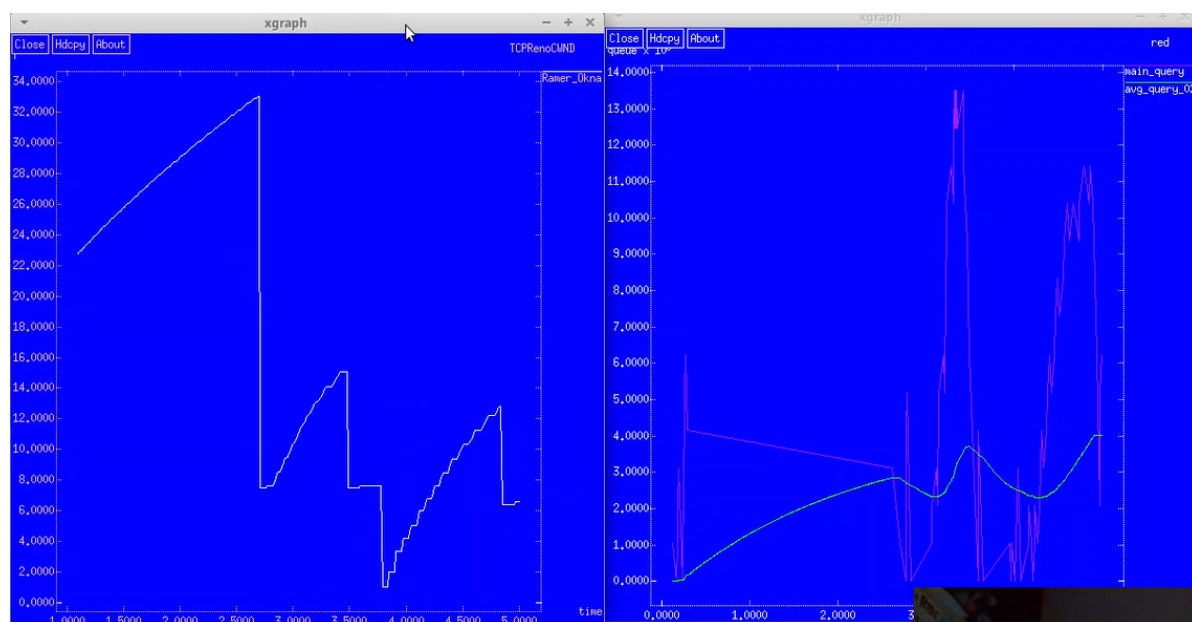
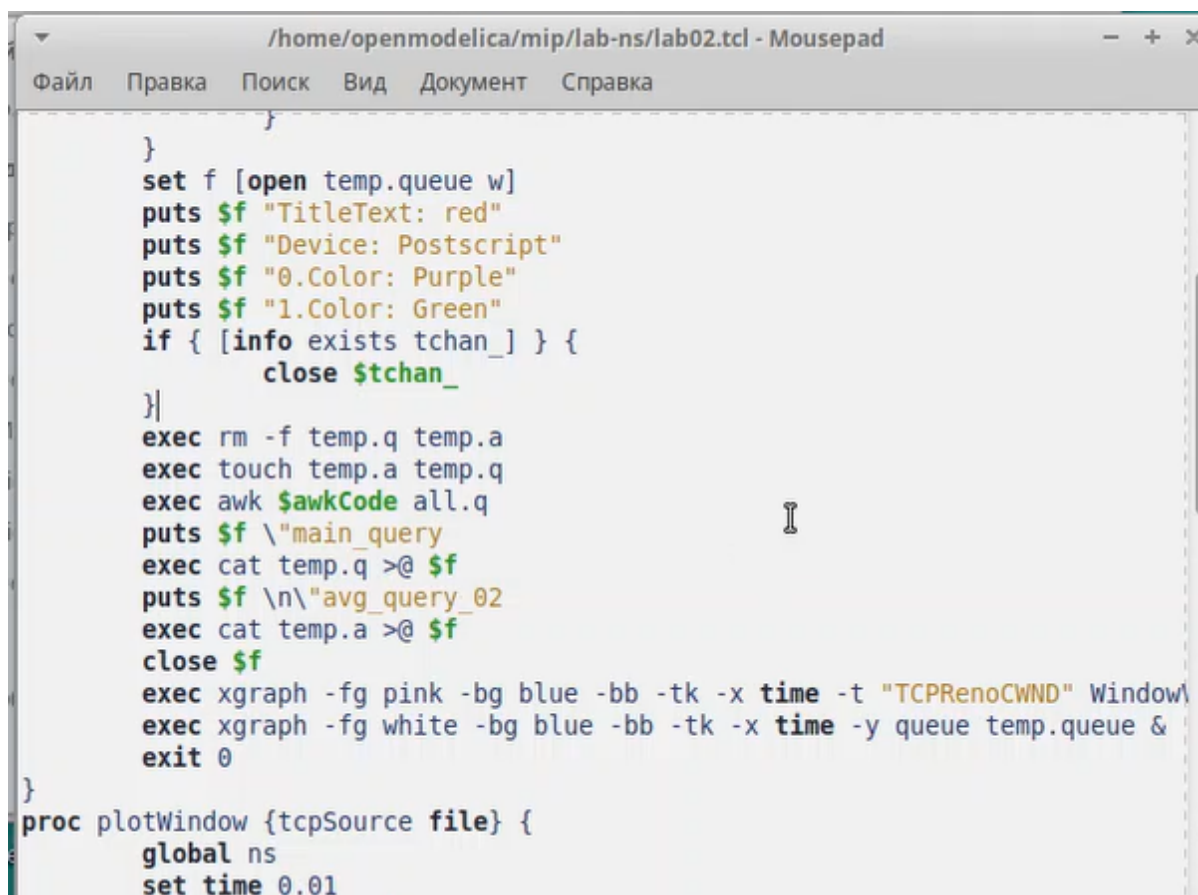


Рис. 3.1: Результат выполнения программы с Reno

Изменила в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas. После сравнения результатов получила, что тип протокола TCP Reno и NewReno похожи между собой, перезагрузка происходит после того, как потерялись некоторые пакеты. у типа Vegas перезагрузка происходит перед тем, как потеряются пакеты. Далее внесла изменения при отображении окон с графиками (изменила цвет фона, цвет траекторий, подписи к осям, подпись траектории в

легенде)(рис.3.2), (рис.3.3), (рис.3.4), (рис.3.5).



```

}
set f [open temp.queue w]
puts $f "TitleText: red"
puts $f "Device: Postscript"
puts $f "0.Color: Purple"
puts $f "1.Color: Green"
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 "\"main_query
exec cat temp.q >@ $f
puts $f \"\\n\\\"avg_query_02
exec cat temp.a >@ $f
close $f
exec xgraph -fg pink -bg blue -bb -tk -x time -t "TCPReNoCWND" Window
exec xgraph -fg white -bg blue -bb -tk -x time -y queue temp.queue &
exit 0
}
proc plotWindow {tcpSource file} {
    global ns
    set time 0.01

```

Рис. 3.2: изменение в коде


```
/home/openmodelica/mip/lab-ns/lab02.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
$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) $node_(r2) 10Mb 5ms DropTail
set tcp1 [$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]
set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "0.Color: White"
puts $windowVsTime \ "Ramer Okna"
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[$ns link $node_(r1) $node_(r2)] queue-sample-timeout;
set redq [[$ns link $node_(r1) $node_(r2)] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
```

Рис. 3.3: Изменение в коде

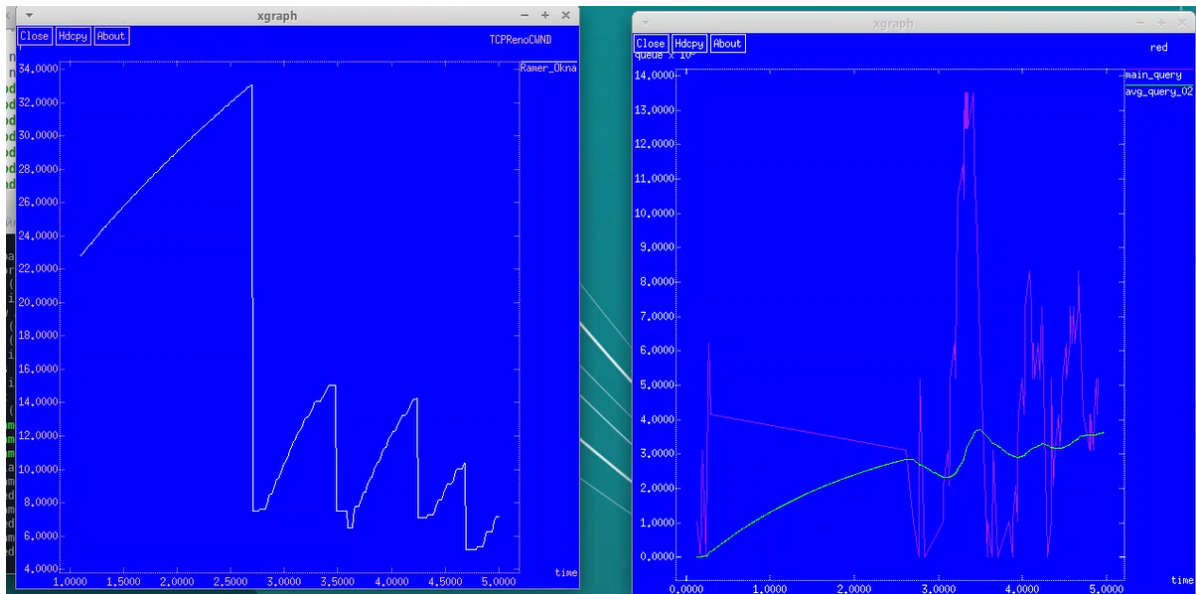


Рис. 3.4: Результат выполнения программы с NewReno



Рис. 3.5: Результат выполнения программы с Vegas

4 Выводы

Исследовала протокол TCP и алгоритм управления очередью RED.

Список литературы