

Лабораторная работа №2

Имитационное моделирование

Волгин Иван Алексеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	17

Список иллюстраций

3.1	Код для модели сети из задания 1	7
3.2	Размер окна	8
3.3	Размер очереди	9
3.4	Изменение типа TCP на Newreno	9
3.5	Размер окна	10
3.6	Размер очереди	11
3.7	Изменение типа TCP на Vegas	11
3.8	Размер окна	12
3.9	Размер очереди	13
3.10	Код для изменения оформления	14
3.11	Вид графика с размером окна	15
3.12	Вид графика с размером очереди	16

1 Цель работы

Целью работы является исследовать протокол TCP и алгоритм управления очередью RED.

2 Задание

- Создать пример сети с дисциплиной RED
- Выполнить дополнительное упражнение

3 Выполнение лабораторной работы

1. В первом задании нужно построить пример сети с дисциплиной RED. 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. Для реализации этой модели написал следующий код (рис. 3.1).

```

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

$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/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]

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;

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"
$ns at 10 "finish"

$ns run

```

Рис. 3.1: Код для модели сети из задания 1

В итоге получил следующие два графика (рис. 3.2) (рис. 3.3). На первом изображен размер окна. Можно заметить, что пиковое значение находится в районе 33, а основной диапазон значений от 1 до 14. На втором графике мы видим размер очереди. Пиковое значение равняется 11, также есть еще пара пиков на значениях 11 и 9, но основной диапазон от 0 до 7.

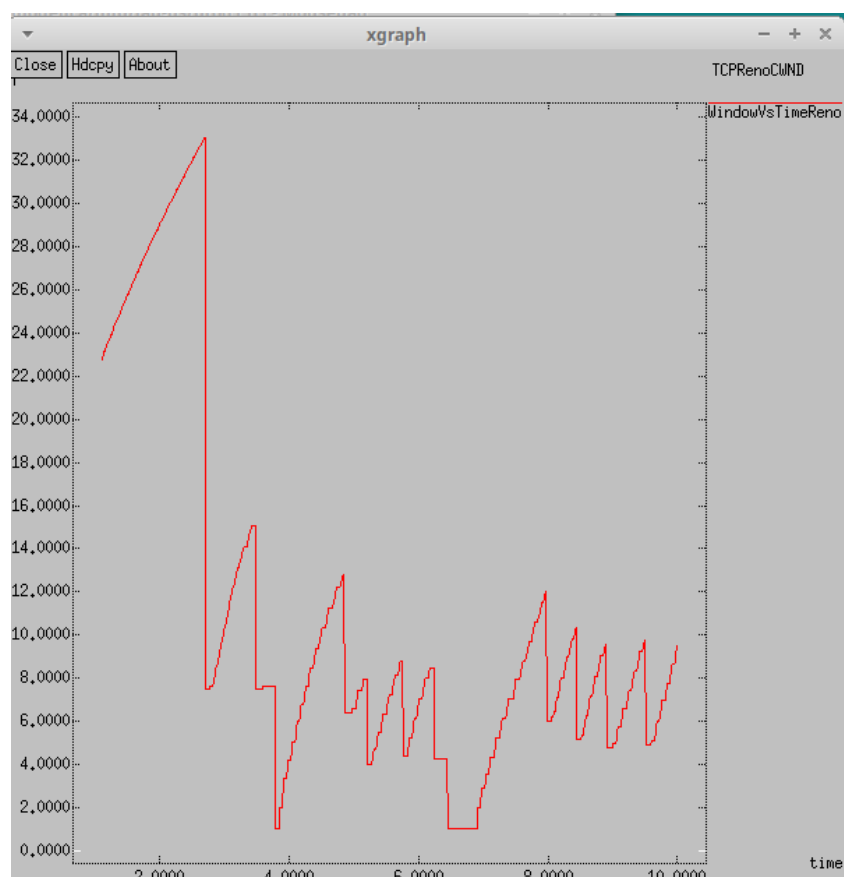


Рис. 3.2: Размер окна

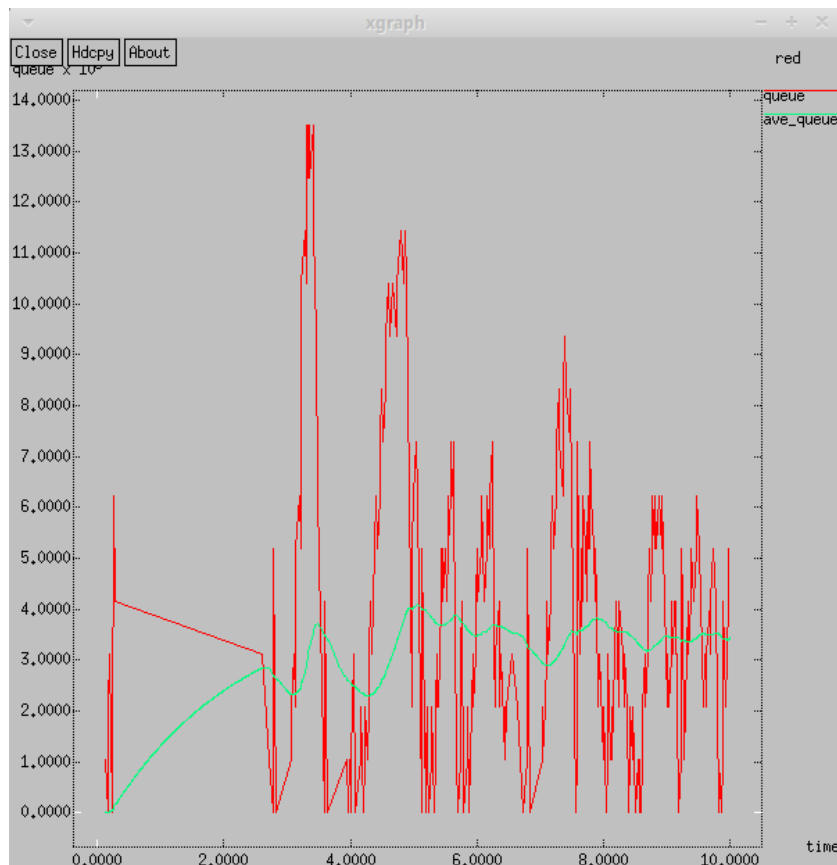


Рис. 3.3: Размер очереди

2. Далее я приступил к выполнению дополнительного упражнения. Задачи: – 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).

В первом пункте нужно было сначала изменить тип протокола TCP на Newreno, что я и сделал (рис. 3.4).

```
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. 3.4: Изменение типа TCP на Newreno

Получил следующие результаты (рис. 3.5) (рис. 3.6). На графике размера окна мы видим, что один пик на значении 33, а затем колебания в диапазоне от 4 до 14. На втором графике, который показывает размер очереди, можно заметить один (в отличие от прошлого Reno) пик на значении 13 и далее однородные колебания в диапазоне 0 - 8.

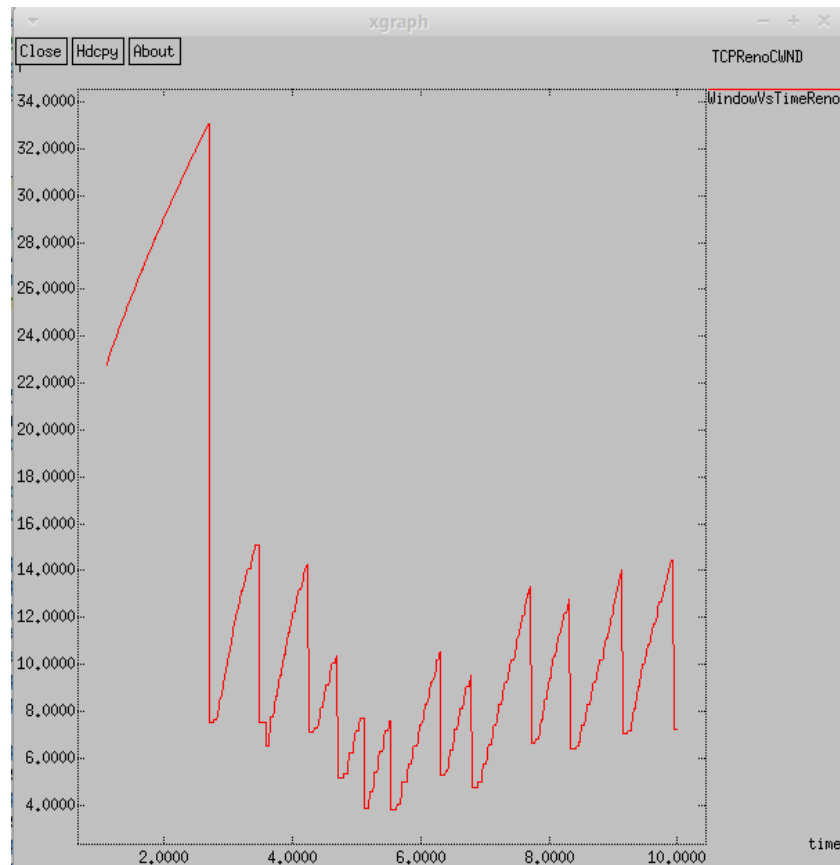


Рис. 3.5: Размер окна

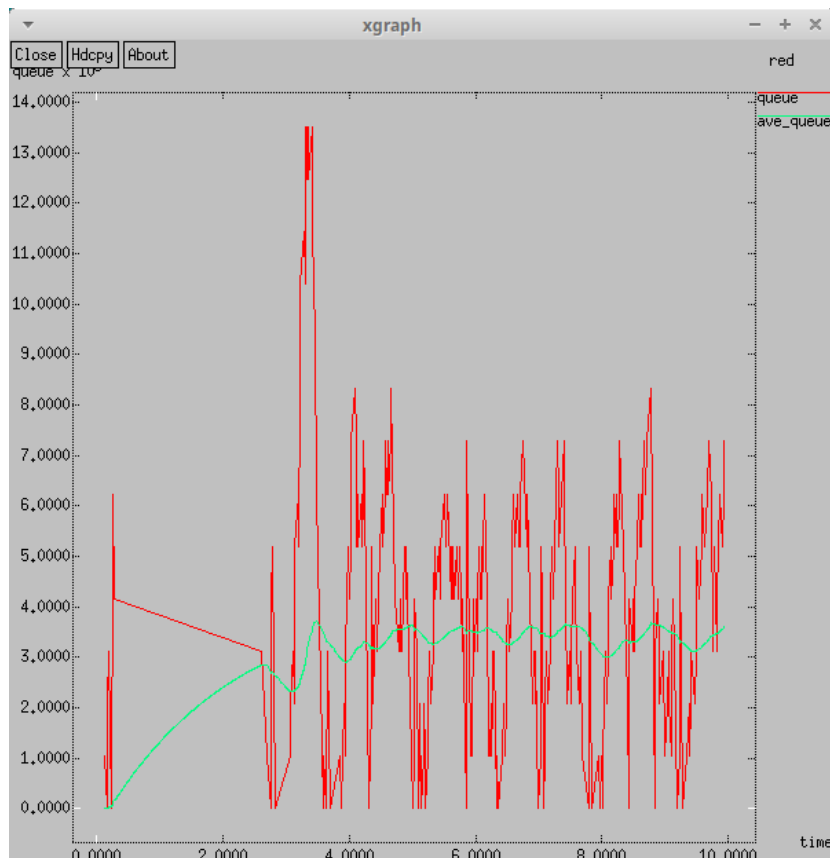


Рис. 3.6: Размер очереди

Далее нужно было поменять тип протокола TCP уже на Vegas. В коде это выглядело так (рис. 3.7)

```
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. 3.7: Изменение типа TCP на Vegas

Я снова получил два графика, которые уже сильно отличаются от двух предыдущих вариантов (рис. 3.8) (рис. 3.9). На первом графике мы видим частые кратковременные пики и периоды по несколько секунд, когда значение размера окна не менялось. Второй меньше первого визуально отличается от предшественников. Пик также располагается на значении 13, а основной диапазон значений от 0 до 8.

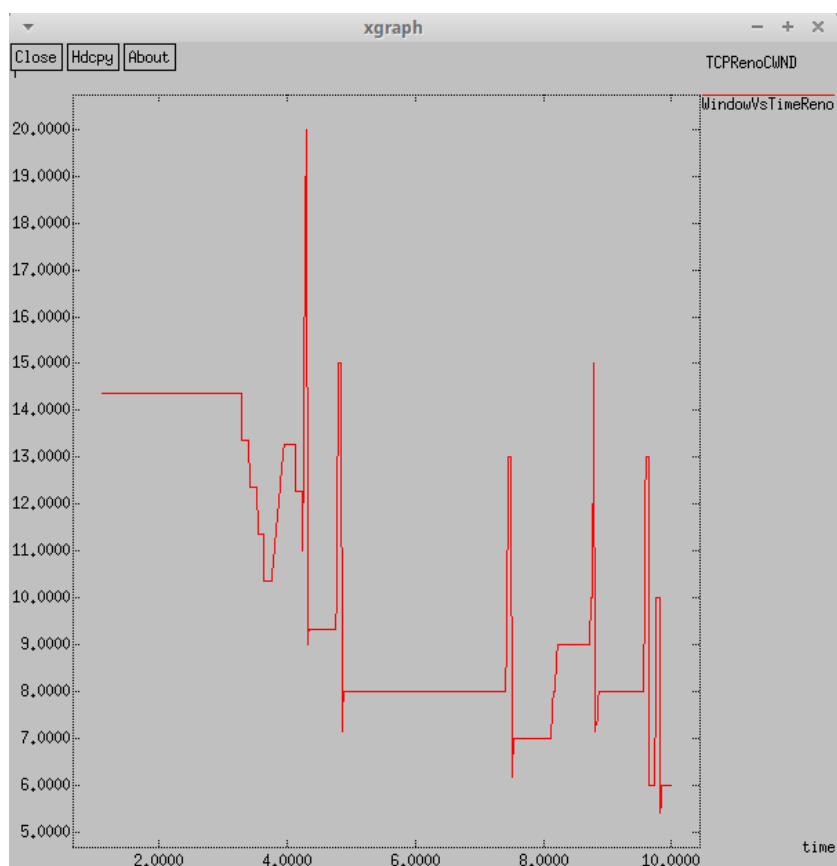


Рис. 3.8: Размер окна

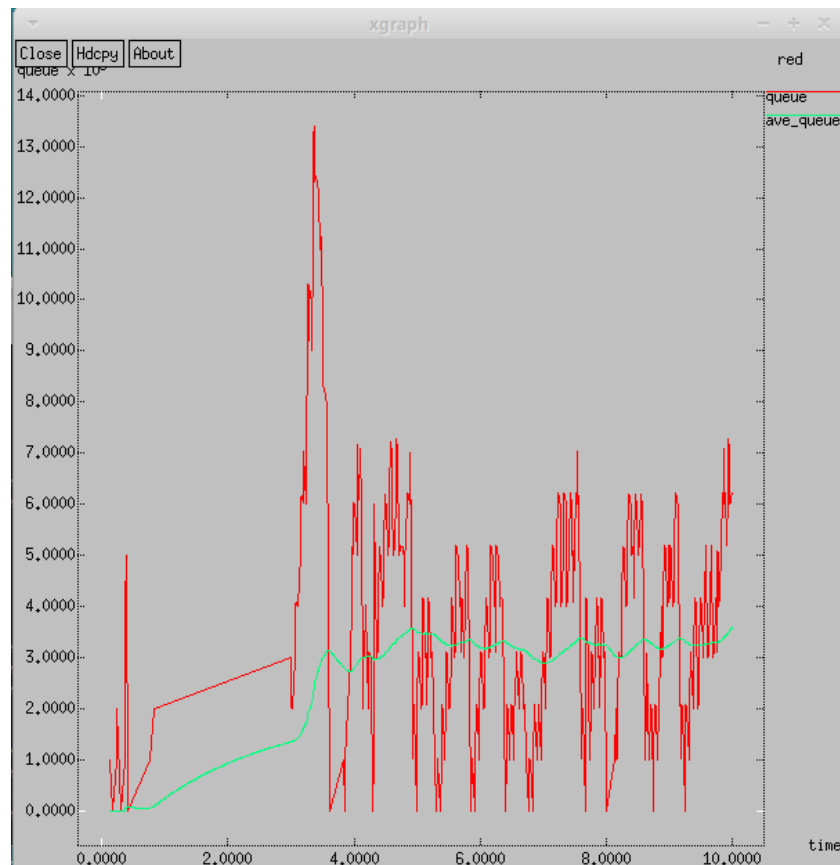


Рис. 3.9: Размер очереди

Вторым пунктом в упражнении было изменение оформления окон с графиками. Нужно было изменить цвет фона, цвет траекторий, подписи к осям и подпись траектории в легенде. Я внес изменения в код (рис. 3.10) и получил следующие результаты (рис. 3.11) (рис. 3.12).

```

}
}
set f [open temp.queue w]
puts $f "TitleText: red"
puts $f "Device: Postscript"
puts $f "0.Color: Yellow"
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 "\nQUEUE"
exec cat temp.q >@ $f
puts $f "\n"AVG QUEUE
exec cat temp.a >@ $f
close $f
exec xgraph -fg blue -bg white -bb -tk -x time -t "TCPReNoCWND" WindowVsTimeReno
exec xgraph -fg gray -bg brown -bb -tk -x time -y queue temp.queue &
exit 0
}

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

$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: Black"
puts $windowVsTime \ "SIZE OF WINDOW"
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
$ns link $node_(r1) $node_(r2) queue-sample-timeout

```

Рис. 3.10: Код для изменения оформления

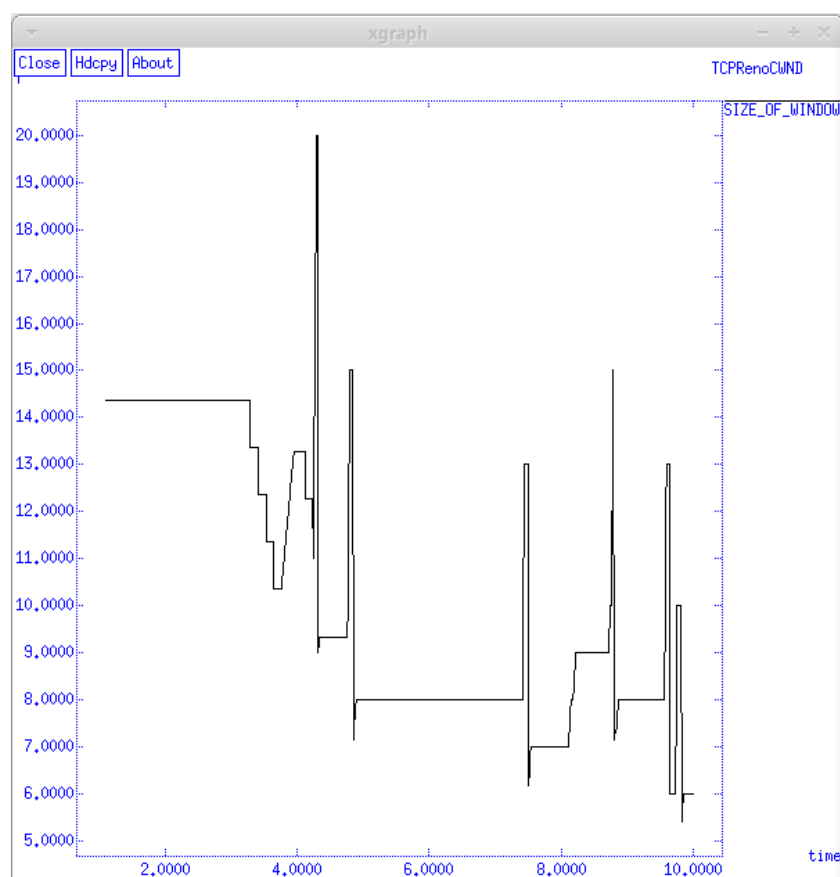


Рис. 3.11: Вид графика с размером окна

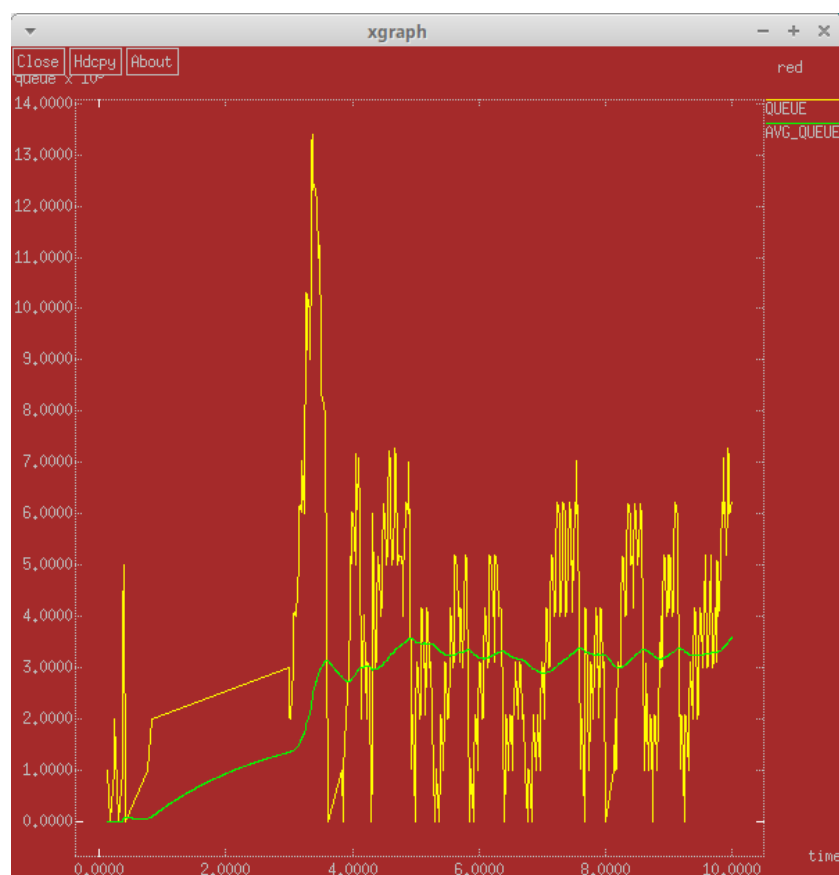


Рис. 3.12: Вид графика с размером очереди

4 Выводы

В ходе выполнения лабораторной работы я исследовал прокол ТСР и алгоритм управления очередью RED