

Лабораторная работа №2

Исследование протокола TCP и алгоритма управления очередью RED

Алиева Милена Арифовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	17

Список иллюстраций

3.1	Редактирование файла	8
3.2	График изменения ТСП-окна	9
3.3	График изменения длины очереди и средней длины очереди . . .	10
3.4	Меняем тип Reno на NewReno	10
3.5	График изменения ТСП-окна	11
3.6	График изменения длины очереди и средней длины очереди . . .	12
3.7	Меняем тип Reno на Vegas	12
3.8	График изменения ТСП-окна	13
3.9	График изменения длины очереди и средней длины очереди . . .	14
3.10	Изменение цвета	14
3.11	Изменение цвета	14
3.12	График изменения ТСП-окна	15
3.13	График изменения длины очереди и средней длины очереди . . .	16

Список таблиц

1 Цель работы

Исследовать протокол TCP и алгоритм управления очередью RED.

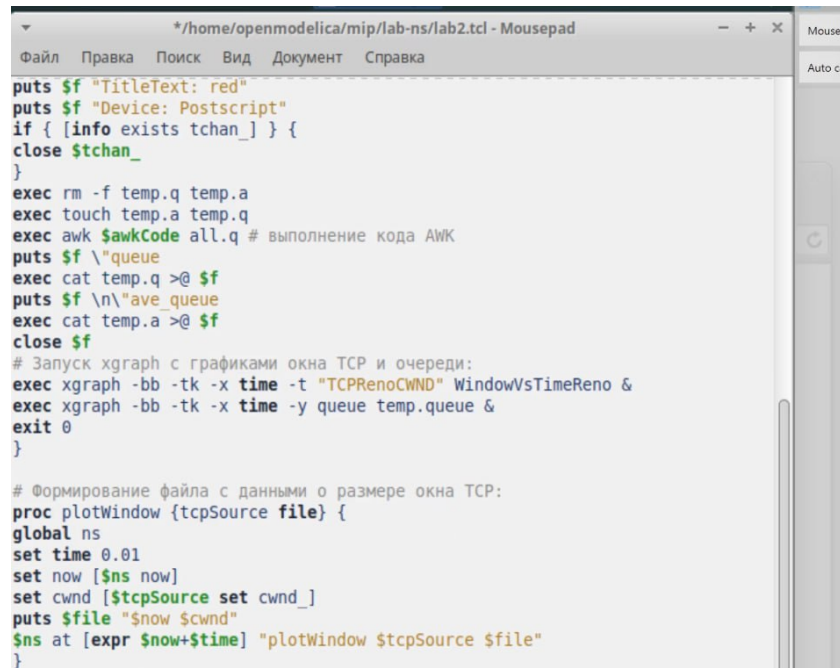
2 Задание

1. Выполнить пример с дисциплиной RED;
2. Изменить в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas. Сравнить и пояснить результаты;
3. Внести изменения при отображении окон с графиками (изменить цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде).

3 Выполнение лабораторной работы

1. Выполним построение сети в соответствии с описанием:

- 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;
- ТСП-источники на узлах s1 и s2 подключаются к ТСП-приёмнику на узле s3; генераторы трафика FTP прикреплены к ТСП-агентам. Теперь разработаем сценарий, реализующий модель согласно описанию, чтобы построить в Xgraph график изменения ТСП-окна, график изменения длины очереди и средней длины очереди.. (рис. 3.1).

A screenshot of a text editor window titled `*/home/openmodelica/mip/lab-ns/lab2.tcl - Mousepad`. The window contains a Tcl script for network simulation. The script includes commands for setting window title and device, checking for information, closing channels, executing system commands like `rm`, `touch`, and `awk`, and using `puts` and `exec` to manage data queues and generate plots with `xgraph`. It also defines a procedure `plotWindow` for periodic updates.

```
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 # выполнение кода AWK
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 "TCPReoCWND" WindowVsTimeReno &
exec xgraph -bb -tk -x time -y queue temp.queue &
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"
}
```

Рис. 3.1: Редактирование файла

2. После запуска кода получаем график изменения TCP-окна (рис. 3.2), а также график изменения длины очереди и средней длины очереди (рис. 3.3).

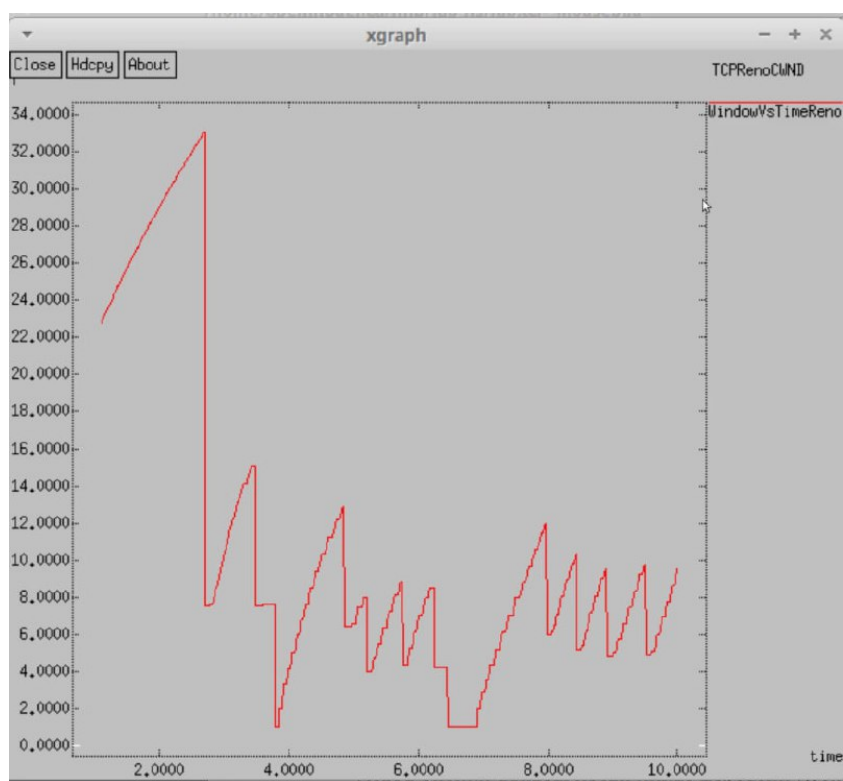


Рис. 3.2: График изменения ТСР-окна

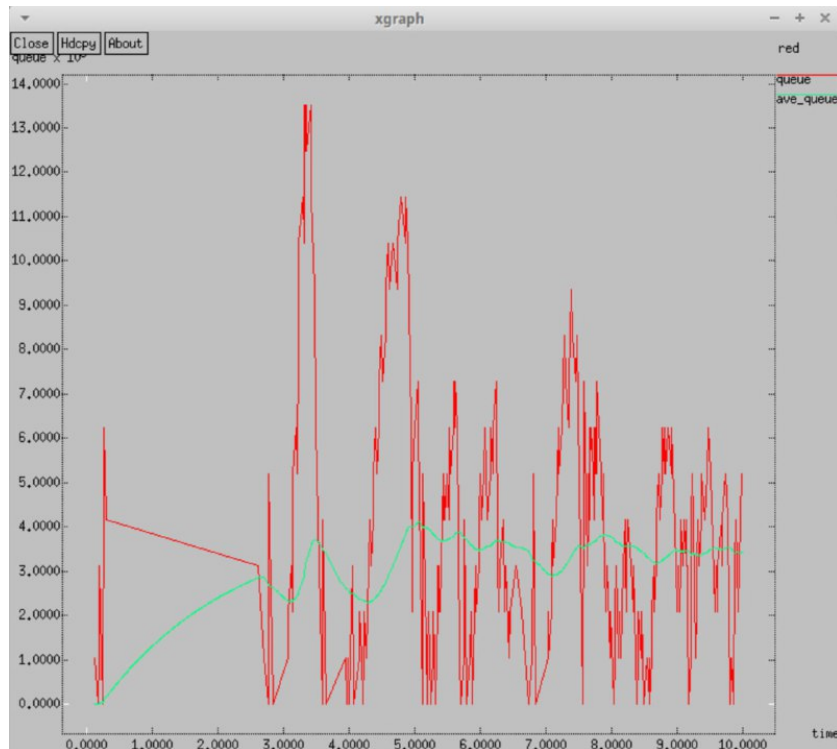


Рис. 3.3: График изменения длины очереди и средней длины очереди

3. Теперь изменим тип Reno на NewReno (рис. 3.4).

```
# 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. 3.4: Меняем тип Reno на NewReno

4. После запуска кода получаем график изменения TCP-окна (рис. 3.5), а также график изменения длины очереди и средней длины очереди (рис. 3.6). Видим, что значение средней длины очереди находится в пределах от 2 до 4, а максимальное значение длины равно 14, как и на предыдущем графике. В обоих алгоритмах размер окна увеличивается до тех пор, пока не произойдёт потеря сегмента.

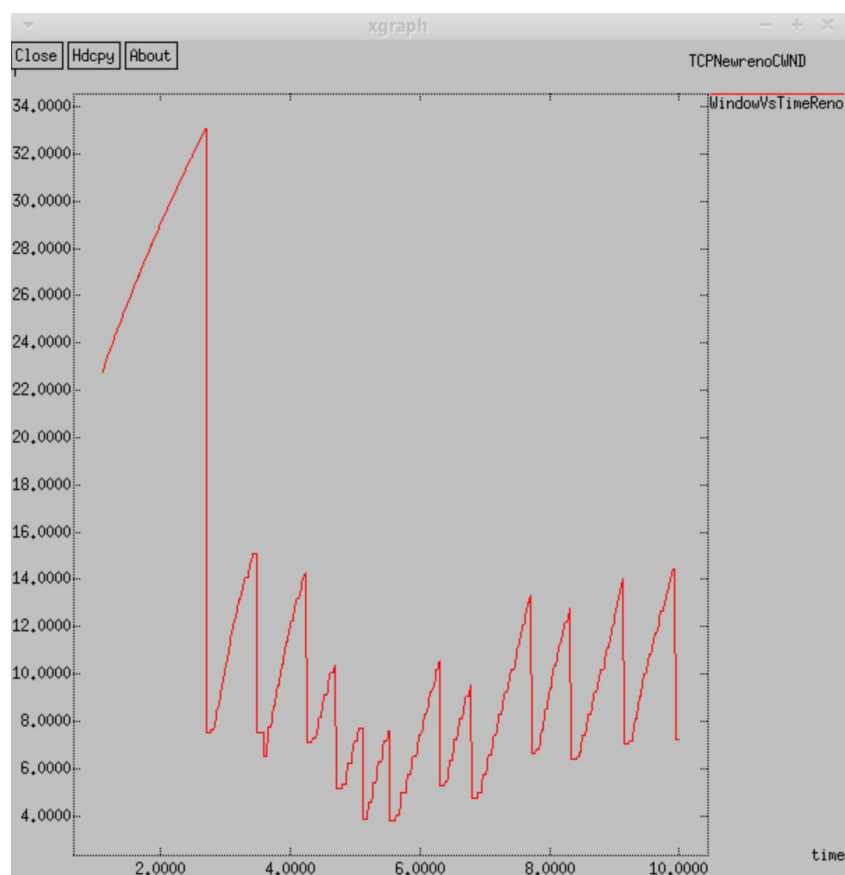


Рис. 3.5: График изменения ТСП-окна

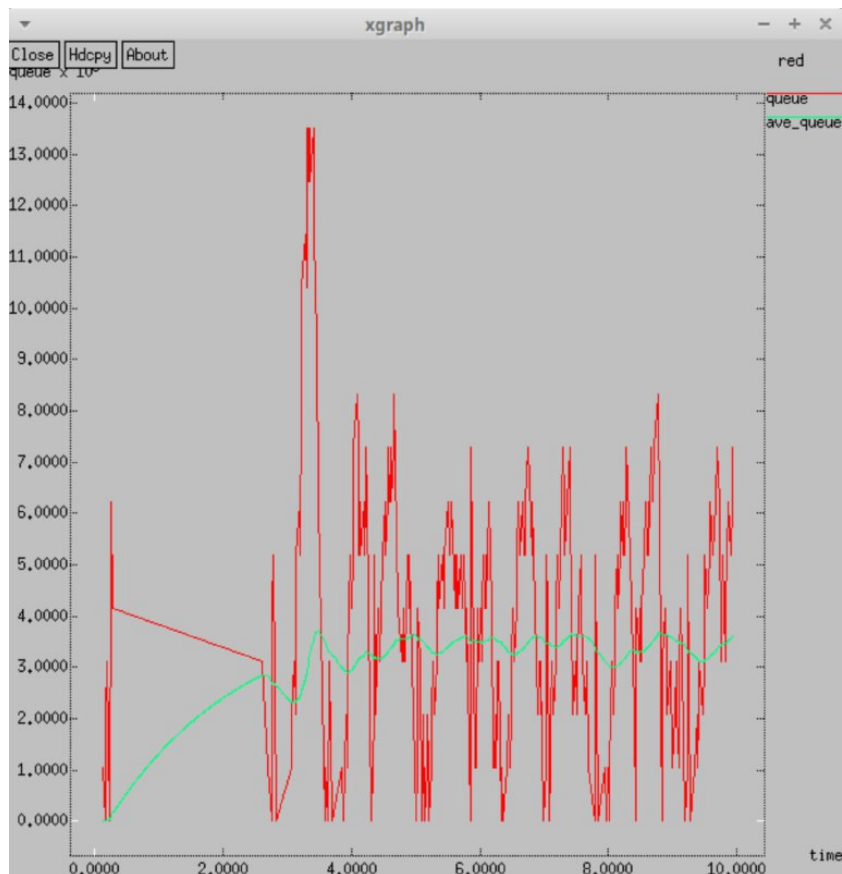


Рис. 3.6: График изменения длины очереди и средней длины очереди

5. Изменим тип Reno на Vegas (рис. 3.7).

```
# 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. 3.7: Меняем тип Reno на Vegas

6. После запуска кода получаем график изменения TCP-окна (рис. 3.8), а также график изменения длины очереди и средней длины очереди (рис. 3.9). Видим, что средняя длина очереди опять находится также в диапазоне от 2 до 4, а максимальная длина достигает значения 14. Довольно сильные отличия можно заметить по графикам динамики размера окна, так как при

Vegas максимальный размер окна составляет 20, а не 34, как в предыдущих графиках. Это происходит по той причине, что TCP Vegas обнаруживает перегрузку в сети до того, как случайно теряется пакет, и мгновенно уменьшается размер окна, получается, что TCP Vegas обрабатывает перегрузку без каких-либо потерь пакета.



Рис. 3.8: График изменения TCP-окна

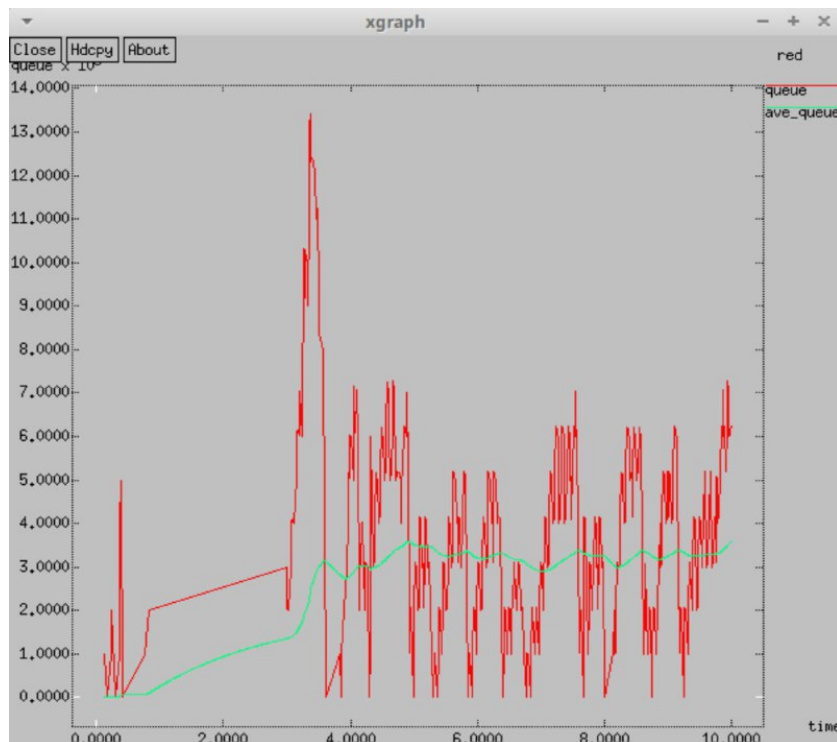


Рис. 3.9: График изменения длины очереди и средней длины очереди

7. Внесем изменения при отображении окон с графиками, изменим цвет фона, цвет траекторий, подписи к осям и подпись траектории в легенде.

```

}
set f [open temp.queue w]
puts $f "TitleText: red"
puts $f "Device: Postscript"
puts $f "0.Color: Blue"
puts $f "1.Color: Pink"
if { [info exists tchan_] } {
close $tchan_
}
exec rm -f temp.q temp.a

```

Рис. 3.10: Изменение цвета

```

close $f
# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -fg pink -bg purple -bb -tk -x time -t "TCP Reno Cwnd" WindowVsTime
exec xgraph -fg white -bg purple -bb -tk -x time -y ochered temp.queue &
exit 0
}
# Формирование файла с данными о размере окна TCP:

```

Рис. 3.11: Изменение цвета

8. После запуска кода получаем график изменения ТСР-окна (рис. 3.12), а также график изменения длины очереди и средней длины очереди (рис. 3.13).

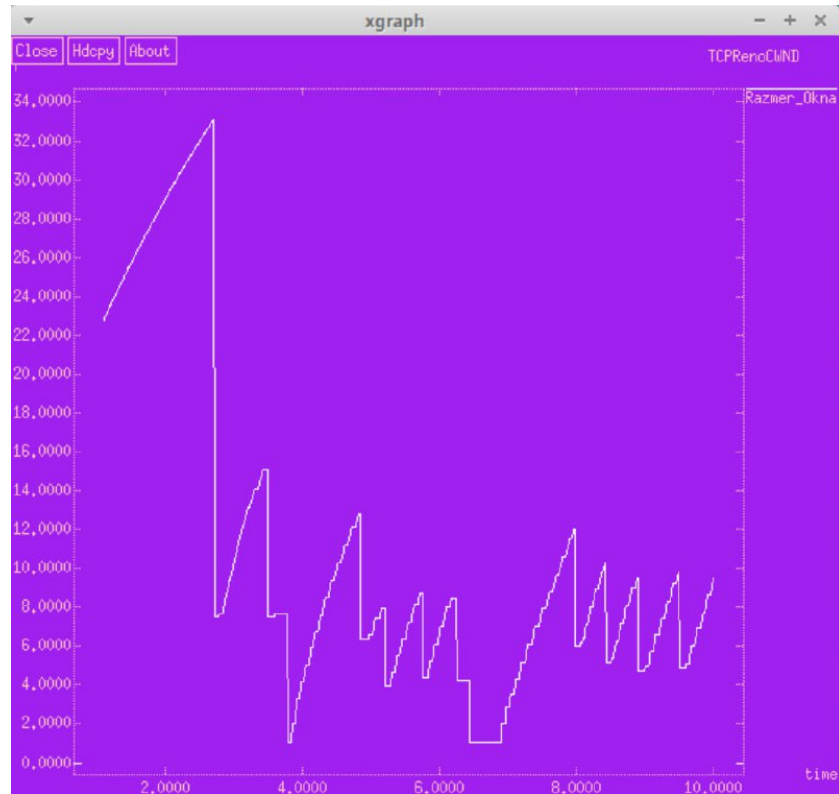


Рис. 3.12: График изменения ТСР-окна



Рис. 3.13: График изменения длины очереди и средней длины очереди

4 Выводы

В процессе выполнения данной лабораторной работы я исследовала протокол TCP и алгоритм управления очередью RED.