

# **Отчёт по лабораторной работе №2**

**Исследование протокола TCP и алгоритма управления  
очередью RED**

Гэинэ Андрей НФИбд-02-22

# Содержание

Цель работы	5
Задание	6
Выполнение лабораторной работы	7
Выводы	14

# Список иллюстраций

1	График изменения TCP-окна, график изменения длины очереди .	7
2	Отображение графиков для Reno . . . . .	8
3	Изменение протокола TCP . . . . .	8
4	Отображение графиков для NewReno 1 . . . . .	9
5	Отображение графиков для NewReno 2 . . . . .	10
6	Изменение протокола TCP . . . . .	10
7	Отображение графиков для Vegas 1 . . . . .	11
8	Отображение графиков для Vegas 2 . . . . .	12
9	Изменение графиков . . . . .	12
10	Видоизмененные графики . . . . .	13

## Список таблиц

# Цель работы

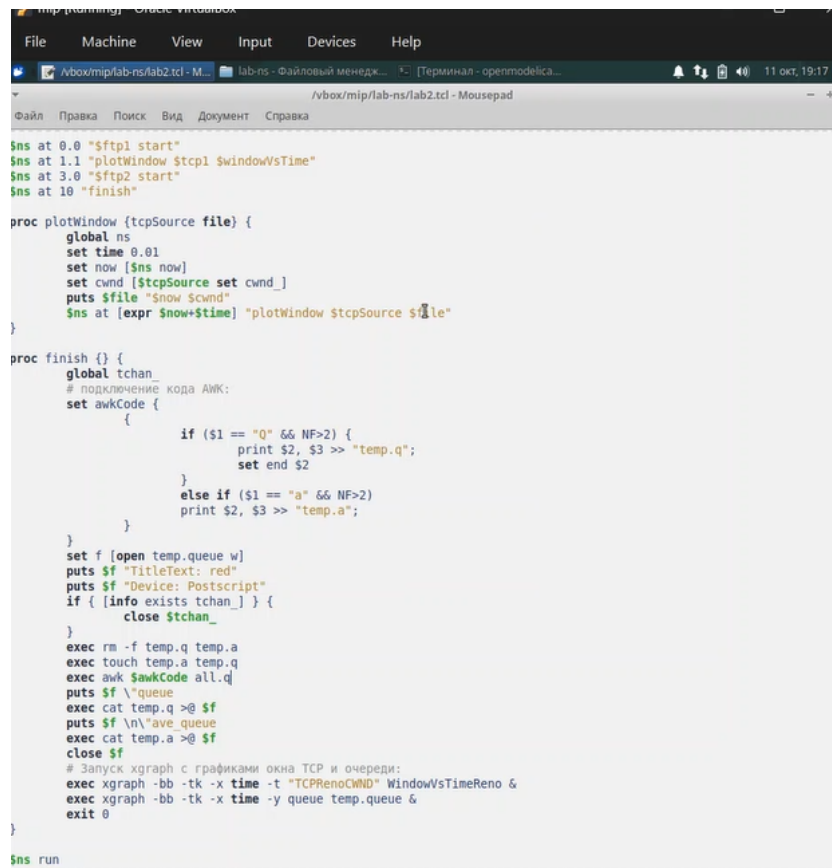
Исследовать протокол TCP и алгоритм управления очередью RED.

# Задание

1. Выполнить пример с дисциплиной RED.
2. Изменить протокол TCP с Reno на NewReno, Vegas. Пояснить результаты.
3. Внести изменения в отображаемые графики.

# Выполнение лабораторной работы

Разработал сценарий, реализующий модель согласно рис. 2.4, построил в Xgraph график изменения TCP-окна, график изменения длины очереди и средней длины очереди. (рис. [-@fig:001])



```
$ns at 0.0 "$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_]
    puts $file "$now cwnd"
    $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

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 "\nqueue"
    exec cat temp.q >@ $f
    puts $f "\nave_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. 1: График изменения TCP-окна, график изменения длины очереди

Отобразил графики, запустив программу. Средняя длина очереди находится в диапазоне от 2 до 4. Макс. длина достигает значения 14. (рис. [-@fig:002])

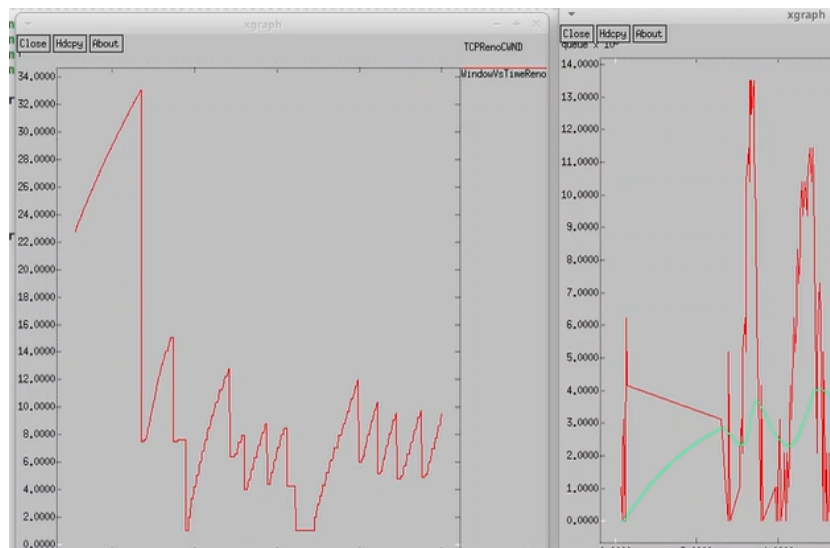


Рис. 2: Отображение графиков для Reno

Изменил тип с Reno на NewReno (рис. [-@fig:003])

```

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

```

Рис. 3: Изменение протокола TCP

Отобразил графики, запустив программу. Значения длины очереди, макс. длины совпадает с предыдущими значениями. В обоих случаях окна увеличиваются до тех пор, пока не произойдет потеря сегмента (рис. [-@fig:004])



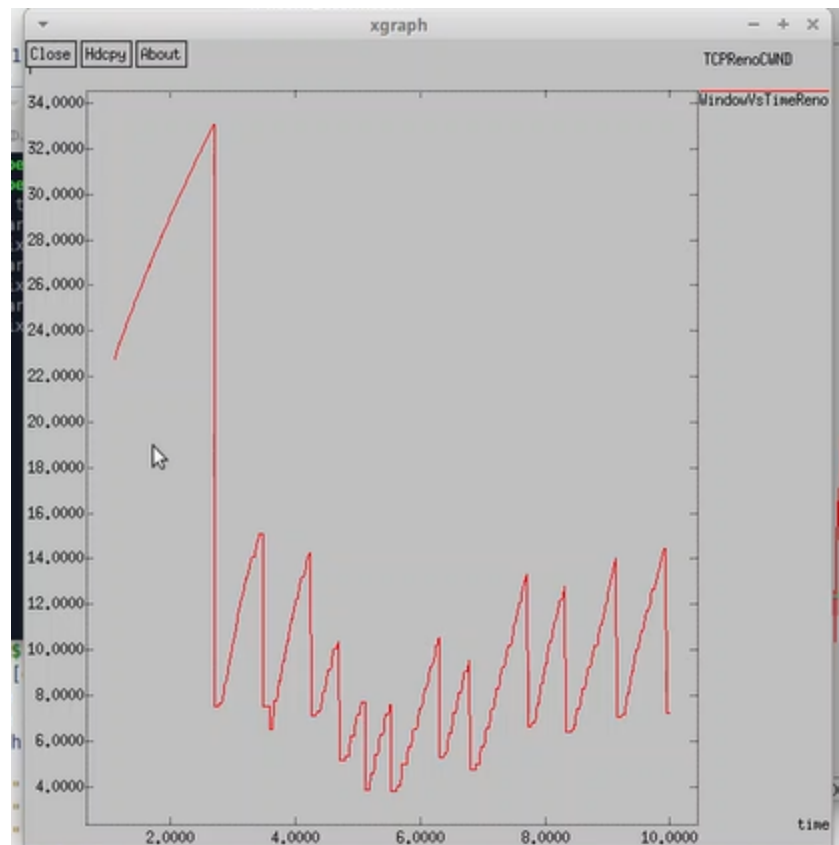


Рис. 4: Отображение графиков для NewReno 1

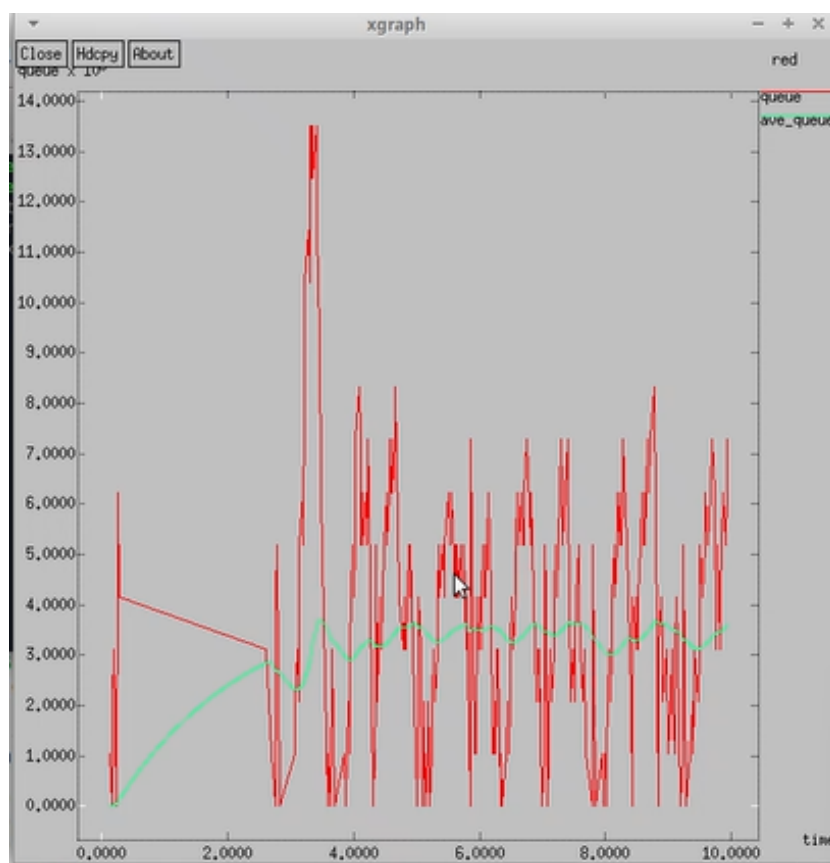


Рис. 5: Отображение графиков для NewReno 2

Изменил тип с Reno на Vegas (рис. [-@fig:005])

```
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. 6: Изменение протокола TCP

Отобразил графики, запустив программу. Видно, что при Vegas макс. размер окна составляет 20, а не 34. TCP Vegas обнаруживает перегрузку до того, как теряется пакет, мгновенно уменьшается размер окна. (рис. [-@fig:006])

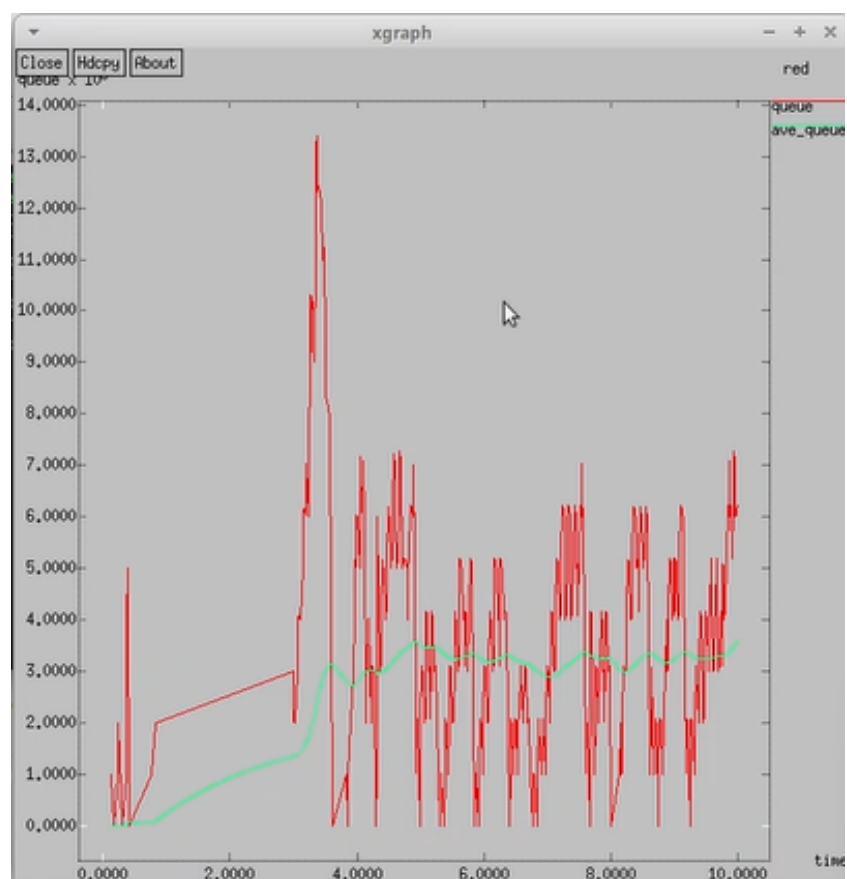


Рис. 7: Отображение графиков для Vegas 1



Рис. 8: Отображение графиков для Vegas 2

Внес изменения в код программы. Поменял цвет фона, траекторий , подписей.  
(рис. [-@fig:007])

```
puts $f "Device: Postscript"
puts $f "Color 1: Orange"
puts $f "Color 2: Pink"
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 "\"AverageQueue"
exec cat temp.a >@ $f
close $f
# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -bb -tk -fg orange -bg pink -x time -t "TCP Reno Cwnd" WindowVsTimeReno &
exec xgraph -bb -tk -fg orange -bg pink -x time -y queue temp.queue &
exit 0
```

Рис. 9: Изменение графиков

Отобразил новые графики, запустив программу. (рис. [-@fig:008])

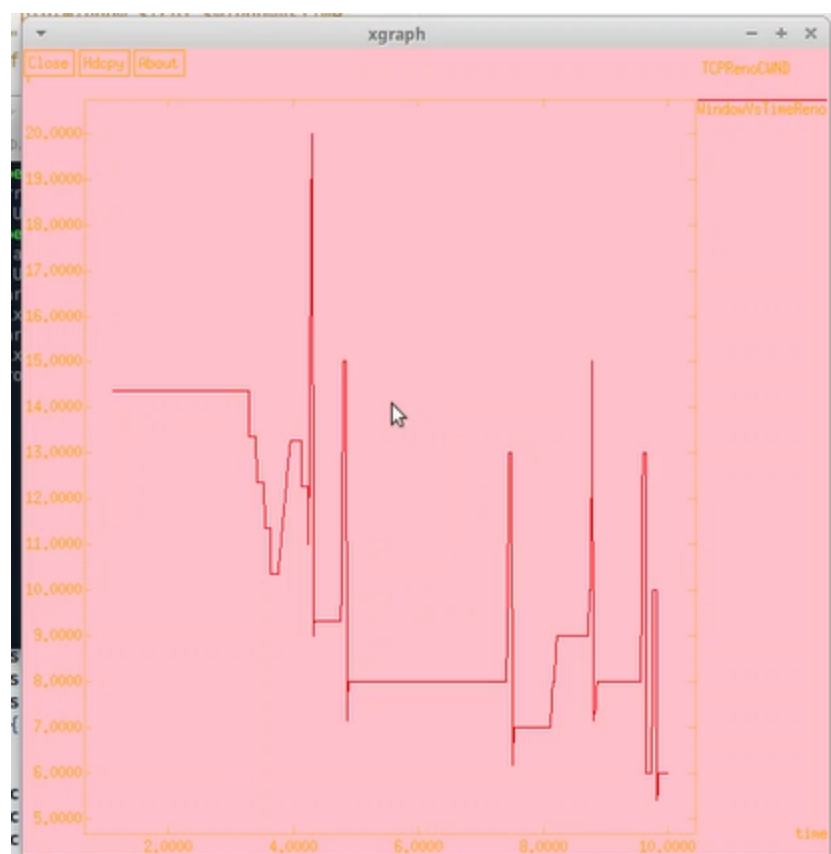


Рис. 10: Видоизмененные графики

# Выводы

Исследовал протокол TCP и алгоритм управления очередью RED.