

Лабораторная Работа №2

Исследование протокола TCP и алгоритма управления очередью RED

Козлов В.П.

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

- Козлов Всеволод Павлович
- НФИбд-02-22
- Российский университет дружбы народов
- [1132226428@pfur.ru]

Выполнение лабораторной работы

Исследовать протокол TCP и алгоритм управления очередью RED.

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

Разработал сценарий, реализующий модель согласно рис. 2.4, построил в Xgraph график изменения ТСР-окна, график изменения длины очереди

и средней длины очереди

```
Терминал - openmodelica@vpkozlov:~/mip/lab-ns
Файл Правка Вид Терминал Вкладки Справка
GNU nano 2.9.3 example8.tcl

global ns
set time 0.01
set now [ns now]
set cwnd [stepSource set cwnd]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow stepSource $file"
}

# Parameters
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]

# Connections
$ns duplex-link $node_($i) $node_r1 10Mb 2ms DropTail
$ns duplex-link $node_($i) $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 $node_($i) $node_r2 10Mb 4ms DropTail
$ns duplex-link $node_($i) $node_r2 10Mb 5ms DropTail

# Sources & Sinks
set tcp1 [$ns create-connection TCP/Reno $node_($i) TCPSink $node_($i) 0]
$tcp1 set window 15
set tcp2 [$ns create-connection TCP/Reno $node_($i) TCPSink $node_($i) 1]
$tcp2 set window 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

# Monitors
$ns monitor $ns TCP
set windowVsTime [open WindowVsTimeReno w]
set qmon [$ns monitor-queue $node_r1 $node_r2 [open qn.out w] 0.1]
$ns link $node_r1 $node_r2 [queue-sample-timeout]

# Traces
set redq [$ns link $node_r1 $node_r2] queue
set tchan [open all.q w]
$redq trace curq_
$redq trace ave
$redq attach $tchan

# Events
# Put events for planning events, which will start
# the procedure finish after 5 s after the start of the simulation
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"

# End of script
```

Отобразил графики, запустив программу. Средняя длина очереди находится в диапазоне от 2 до 4. Макс. длина достигает значения 14

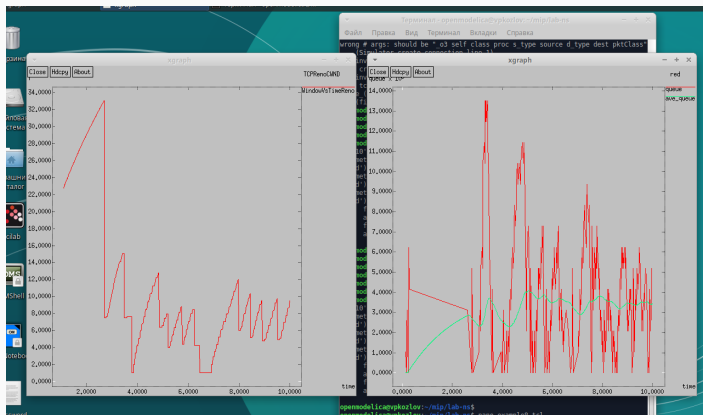


Figure 2: Отображение графиков для Reno

Изменил тип с Reno на NewReno

```
$ns duplex-link $node_($4, $node_($2, 10Mb 5ms DropTail
# $node_($4, $node_($2, 10Mb 5ms DropTail
set tcp1 [$ns create-connection TCP/Newreno $node_($1) TCPSink $node_($3) 0]
$tcp1 set window 15
set tcp2 [$ns create-connection TCP/Reno $node_($2) TCPSink $node_($3) 1]
$tcp2 set window 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
```

Figure 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

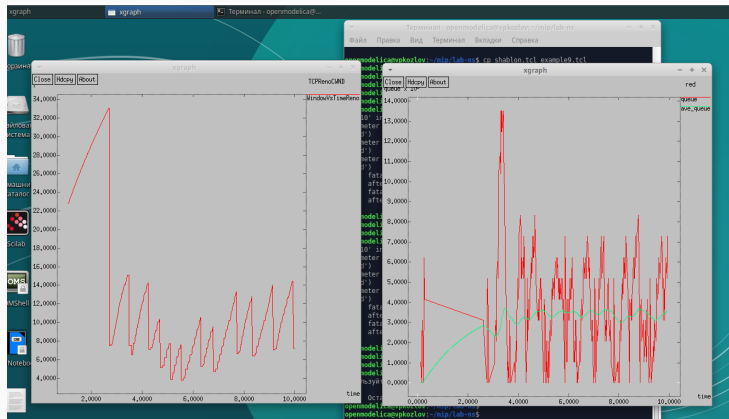


Figure 4: Отображение графиков для NewReno

Изменил тип с Reno на Vegas

```
$ns duplex-link $node_($4) $node_($r2) 10MB $ns DropTail
$ns queue $ns queue-size
set tcp1 [$ns create-connection TCP/Vegas $node_($s1) TCPSink $node_($s3) 0]
step1 set window 15
set tcp2 [$ns create-connection TCP/Reno $node_($s2) TCPSink $node_($s3) 1]
step2 set window 15
set ftp1 [step1 attach-source FTP]
set ftp2 [step2 attach-source FTP]
```

Figure 5: 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

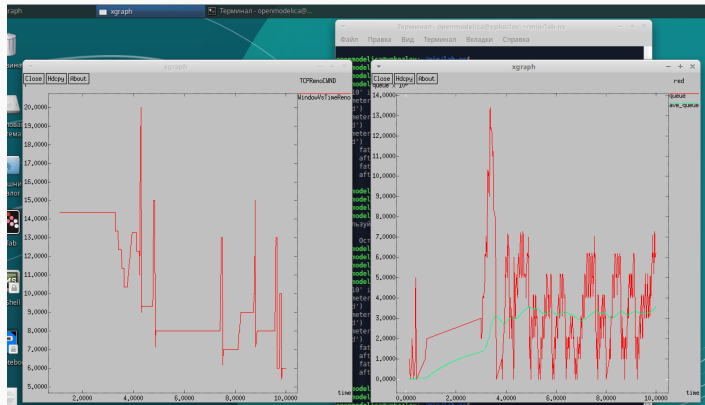
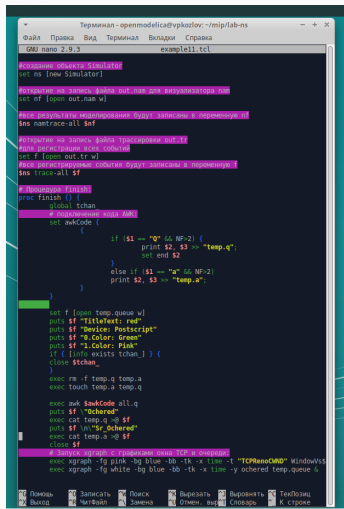


Figure 6: Отображение графиков для Vegas

Внес изменения в код программы. Поменял цвет фона, траекторий , подписей



```
Терминал - openmodelica@vprkzlov: ~/mip/lab-ns
GNU nano 2.9.3 example1.tcl

# Создание объекта Simulator
set ns [new Simulator]

# Открытие на запись файла out.nam для записываемого nam
set nf [open out.nam w]

# Если результаты моделирования будут записаны в переменную x
$ns namtrace-all $nf

# Открытие на запись файла трассировки out.tr
# Для трассировки всех событий
set f [open out.tr w]
# Если зарегистрированные события будут записаны в переменную y
$ns trace-all $f

# Процедура finish
proc finish () {
    global tchan
    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 "Title: text: red"
    puts $f "Device: Postscript"
    puts $f "0.Color: Green"
    puts $f "1.Color: 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 "\033[0m"
    exec cat temp.q > $f
    puts $f "\033[0m"
    exec cat temp.a > $f
    close $f

    # Sanyas xgraph - графический редактор
    exec xgraph -fg pink -bg blue -bb -tk -x time -t "TCP Reno Cwnd" WindowVst
    exec xgraph -fg white -bg blue -bb -tk -x time -y ochered temp.queue &
}
```

Figure 7: Изменение графиков

Отобразил новые графики, запустив программу

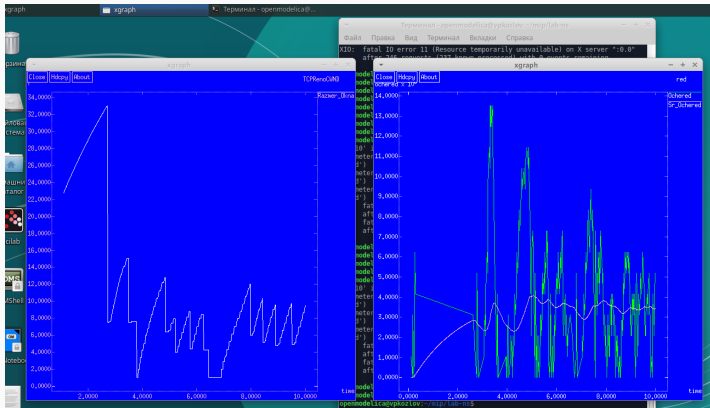


Figure 8: Видоизмененные графики

Исследовал протокол TCP и алгоритм управления очередью RED.