

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

Студент: Шалыгин Георгий Эдуардович

Группа: НФИбд-02-20

Постановка задачи

Изучение протокола TCP и алгоритма управления очередью RED.

Задачи:

- Запустить симуляцию для предложенной в примере сети.
- Изменить в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas. Сравните и поясните результаты.
- Внести изменения при отображении окон с графиками (измените цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде).

Создание шаблона сценария

```
set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

set f [open out.tr w]

$ns trace-all $f

proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

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

set n_(r1) [$ns node]
set n_(r2) [$ns node]

$ns duplex-link $n_(s1) $n_(r1) 10Mb 2ms DropTail
$ns duplex-link $n_(s2) $n_(r1) 10Mb 2ms DropTail
$ns duplex-link $n_(r1) $n_(r2) 1.5Mb 20ms RED
$ns queue-limit $n_(r1) $n_(r2) 25
$ns queue-limit $n_(r2) $n_(r1) 25
$ns duplex-link $n_(s3) $n_(r2) 10Mb 4ms DropTail
$ns duplex-link $n_(s4) $n_(r2) 10Mb 5ms DropTail

set tcp1 [$ns create-connection TCP/Reno $n_(s1) TCPSink $n_(s3) 0]
set tcp2 [$ns create-connection TCP/Reno $n_(s2) TCPSink $n_(s3) 1]
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
```

Создание шаблона сценария

```
set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "0.Color: green"
set qmon [$ns monitor-queue $n_(r1) $n_(r2) [open qm.out w] 0.1];
[$ns link $n_(r1) $n_(r2)] queue-sample-timeout;
```

```
set redq [[$ns link $n_(r1) $n_(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"
```

```
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
    set awkCode {
        {
            if ($1 == "0" && 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"
```

Создание шаблона сценария

```
        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

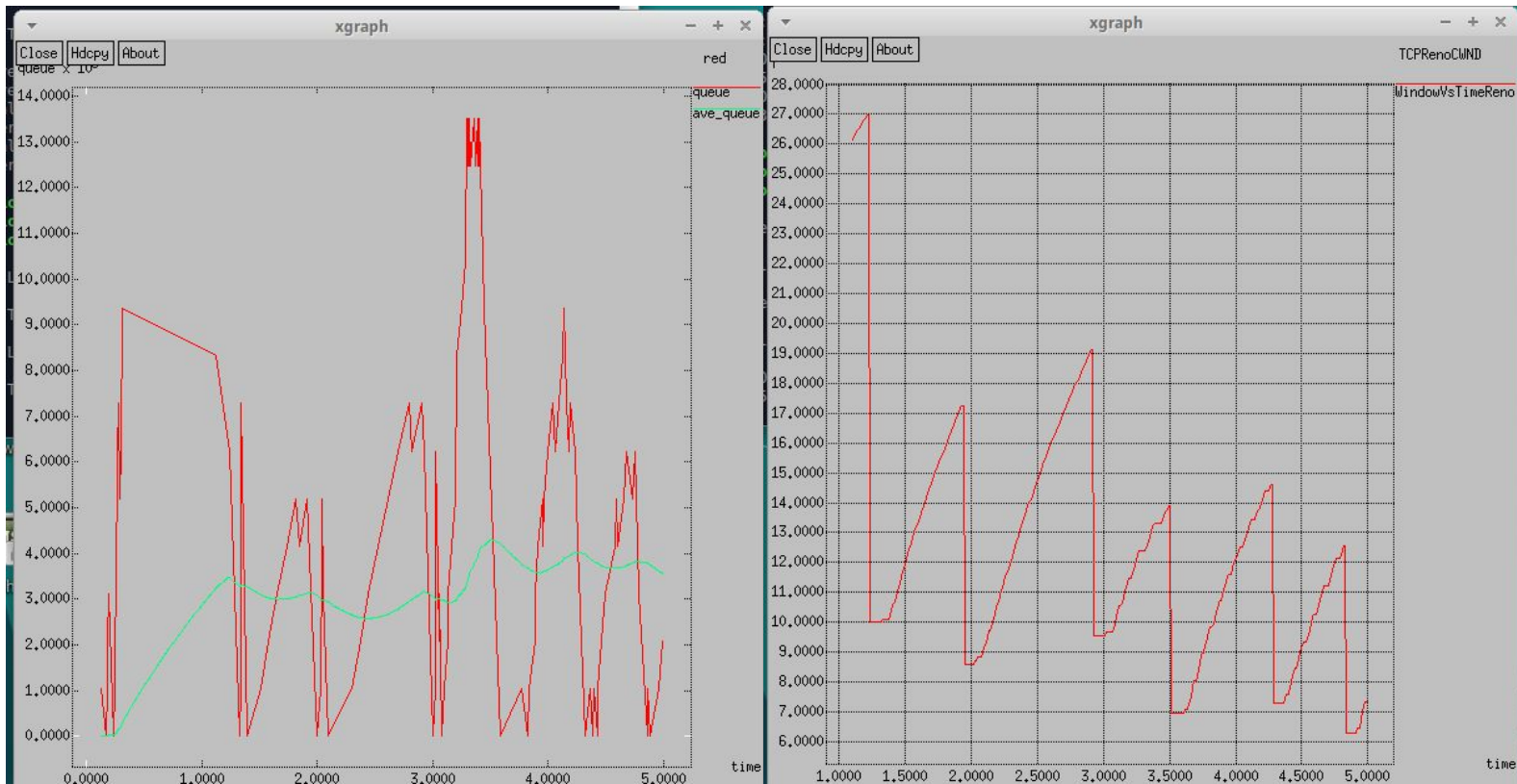
    exec awk $awkCode all.q

    puts $f "\"Queue
    exec cat temp.q >@ $f
    puts $f "\\n\\\"Ave_queue
    exec cat temp.a >@ $f
    close $f

    exec xgraph -bb 0tk -x time "TCPReNoCWND" WindowVsTimeReno &
    exec xgraph -bb -tk -x time temp.queue &
    exit 0
}

$ns at 5.0 "finish"
$ns run
```

Результат моделирования



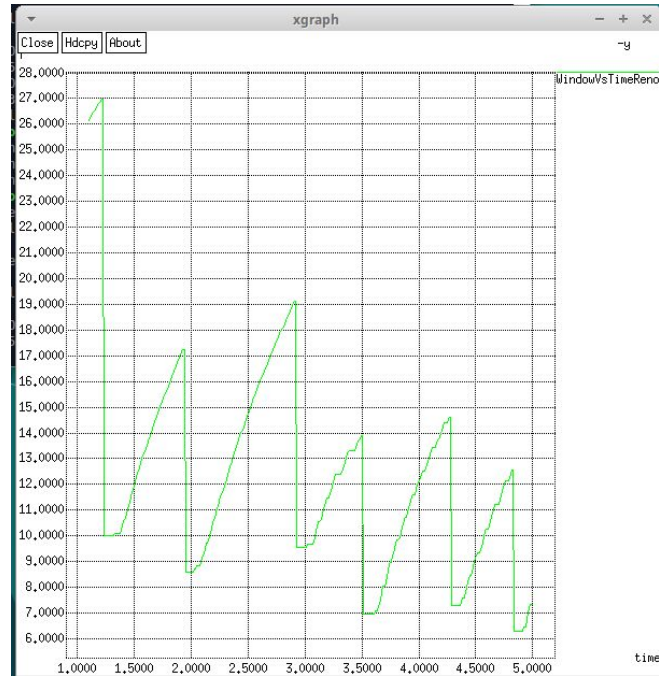
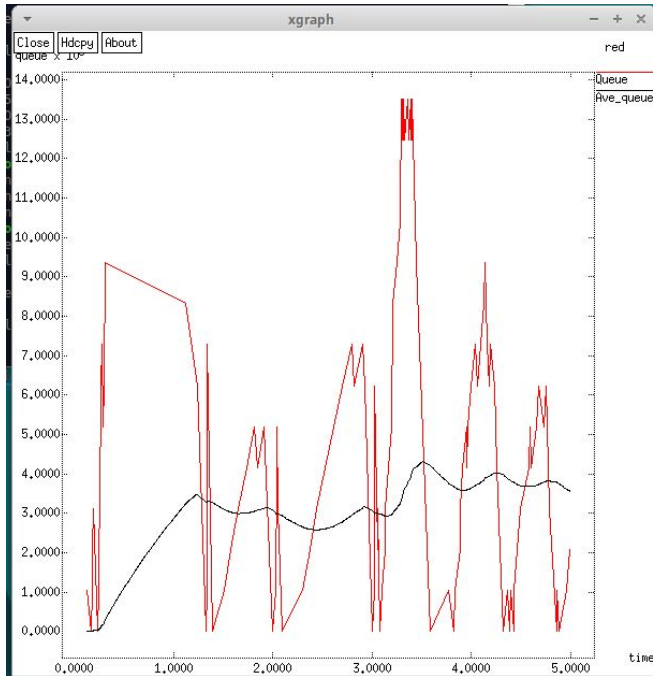
Изменения

```
puts $f "0.Color: red"  
puts $f "1.Color: black"
```

```
exec xgraph -bg white -bb 0tk -x time -t -y Y "TCPRenoCWND" WindowVsTimeReno &  
exec xgraph -bg white -bb -tk -x time -y Queue_cnt -y queue temp.queue &
```

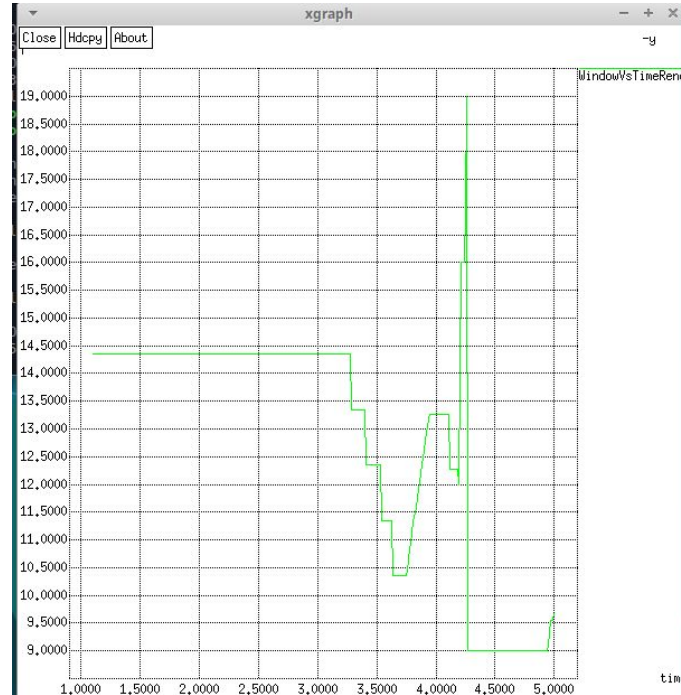
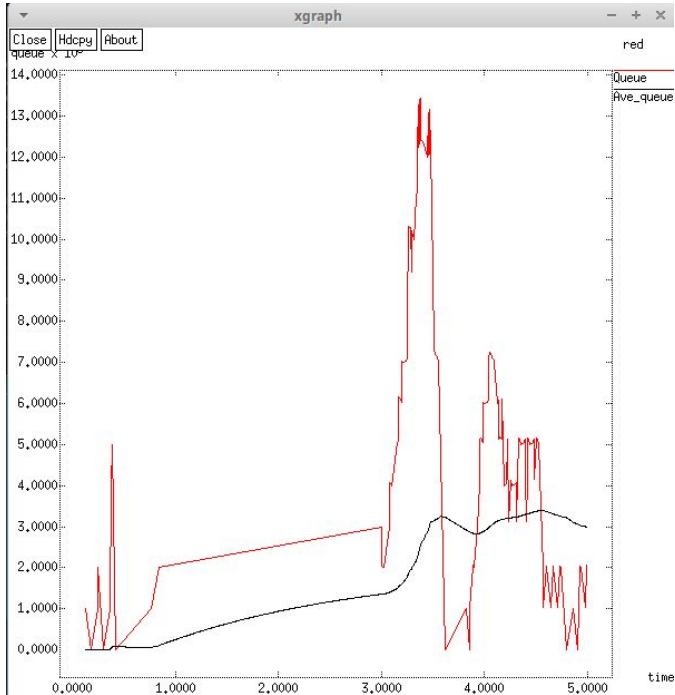
Алгоритм управления очередью NewReno

```
set tcp1 [$ns create-connection TCP/Newreno $n_(s1) TCPSink $n_(s3) 0]  
set tcp2 [$ns create-connection TCP/Reno $n_(s2) TCPSink $n_(s3) 1]  
...  
set $ns1 [$ns1 attach-agent TCP]
```



Алгоритм управления очередью Vegas

```
set tcp1 [$ns create-connection TCP/Vegas $n_(s1) TCPSink $n_(s3) 0]  
set tcp2 [$ns create-connection TCP/Reno $n_(s2) TCPSink $n_(s3) 1]
```



Заключение

В итоге были рассмотрены протоколы tcr и тип управления очередью red. А также средство Xgraph.