

Лабораторной работе №2.

Исследование протокола TCP и алгоритма управления очередью RED

Коне Сирики. НФИбд-01-20

Содержание

1	Цель лабораторной работы:	5
2	Выполнение теорической часть :	6
2.1	Пример задания множества объектов мониторинга:	6
3	Задача лабораторной работы:	8
3.1	Код программы:	8
4	Результаты работы программы	12
5	Выводы	16

Список иллюстраций

[illegible]

Список таблиц

1 Цель лабораторной работы:

Цель работы - познакомиться с работой с исследованием протокола TCP и алгоритма управления очередью RED

2 Выполнение теорической часть :

2.1 Пример задания множества объектов мониторинга:

```
SimpleLink instproc \  
attach-monitors { insnoop outsnoop dropsnoop qmon } {  
$self instvar queue_ head_ snoopIn_ snoopOut_ snoopDrop_  
$self instvar drophead_ qMonitor_  
set snoopIn_ $insnoop  
set snoopOut_ $outsnoop  
set snoopDrop_ $dropsnoop  
$snoopIn_ target $head_  
set head_ $snoopIn_  
$snoopOut_ target [$queue_ target]  
$queue_ target $snoopOut_  
$snoopDrop_ target [$drophead_ target]  
$drophead_ target $snoopDrop_  
$snoopIn_ set-monitor $qmon  
$snoopOut_ set-monitor $qmon  
$snoopDrop_ set-monitor $qmon  
set qMonitor_ $qmon  
}  
SimpleLink instproc init-monitor { ns qtrace sampleInterval} {  
$self instvar qMonitor_ ns_ qtrace_ sampleInterval_
```

```
set ns_ $ns
set qtrace_ $qtrace
set sampleInterval_ $sampleInterval
set qMonitor_ [new QueueMonitor]
$self attach-monitors [new SnoopQueue/In]
[new SnoopQueue/Out] [new SnoopQueue/Drop] $qMonitor_
set bytesInt_ [new Integrator]
$qMonitor_ set-bytes-integrator $bytesInt_
set pktsInt_ [new Integrator]
$qMonitor_ set-pkts-integrator $pktsInt_
return $qMonitor_ }
```

3 Задача лабораторной работы:

1. Измените в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas. Сравните и поясните результаты.
2. Внесите изменения при отображении окон с графиками (измените цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде).

3.1 Код программы:

```
set ns [new Simulator]
set node_(s1) [$ns node]
set node_(s2) [$ns node]
set node_(r1) [$ns node]
set node_(r2) [$ns node]
set node_(s3) [$ns node]
set node_(s4) [$ns node]

# Соединения:
$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

$ns duplex-link-op $node_(s1) $node_(r1) orient right-down
$ns duplex-link-op $node_(s2) $node_(r1) orient right-up
$ns duplex-link-op $node_(r1) $node_(r2) orient right
$ns duplex-link-op $node_(r1) $node_(r2) queuePos 0
$ns duplex-link-op $node_(r2) $node_(r1) queuePos 0
$ns duplex-link-op $node_(s3) $node_(r2) orient left-down
$ns duplex-link-op $node_(s4) $node_(r2) orient left-up

# Агенты и приложения:
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]

# Мониторинг размера окна TCP:
#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;

# Мониторинг очереди:
set redq [[$ns link $node_(r1) $node_(r2)] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_

# Добавление at-событий:
$ns at 0.0 "$ftp1 start"

```

```

# $ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 25 "finish"
proc finish {} {
    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"
    global tchan_
    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 -tk -x time -y queue temp.queue &
exit 0
}
$ns run
```

4 Результаты работы программы

(рис. 4.1).

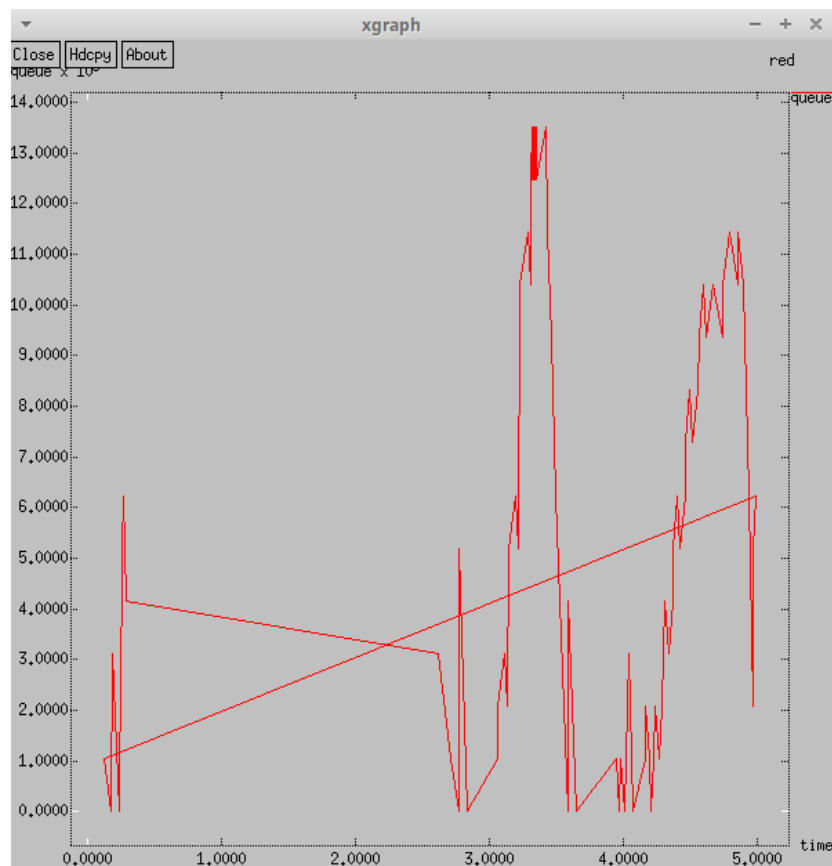


Рис. 4.1: 1

(рис. 4.2).

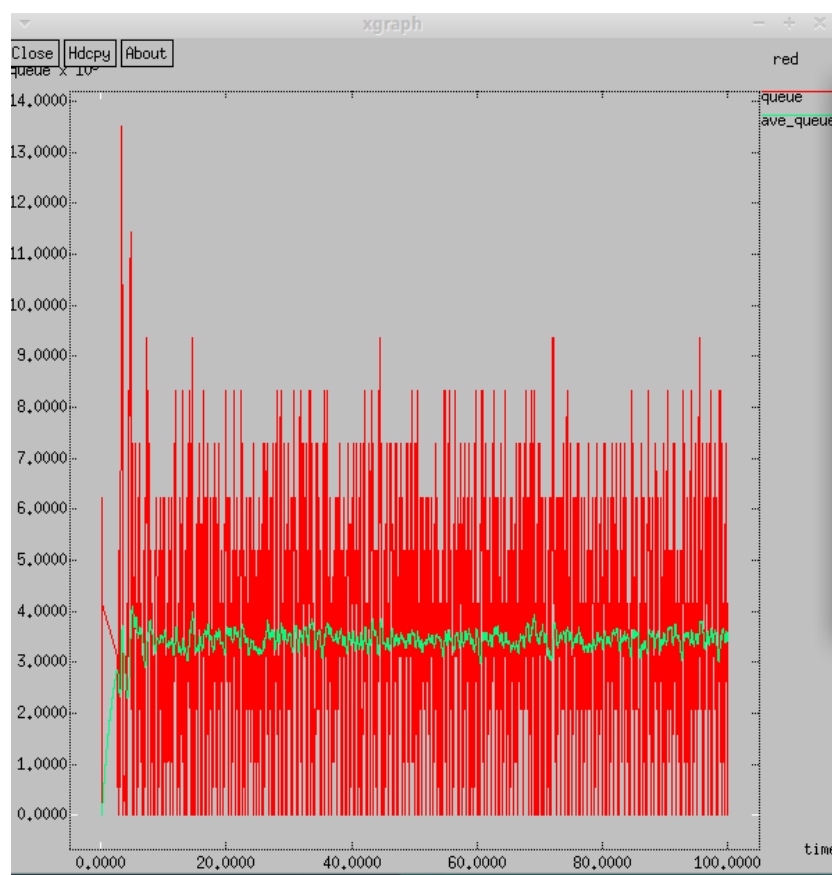


Рис. 4.2: 2

(рис. 4.3).

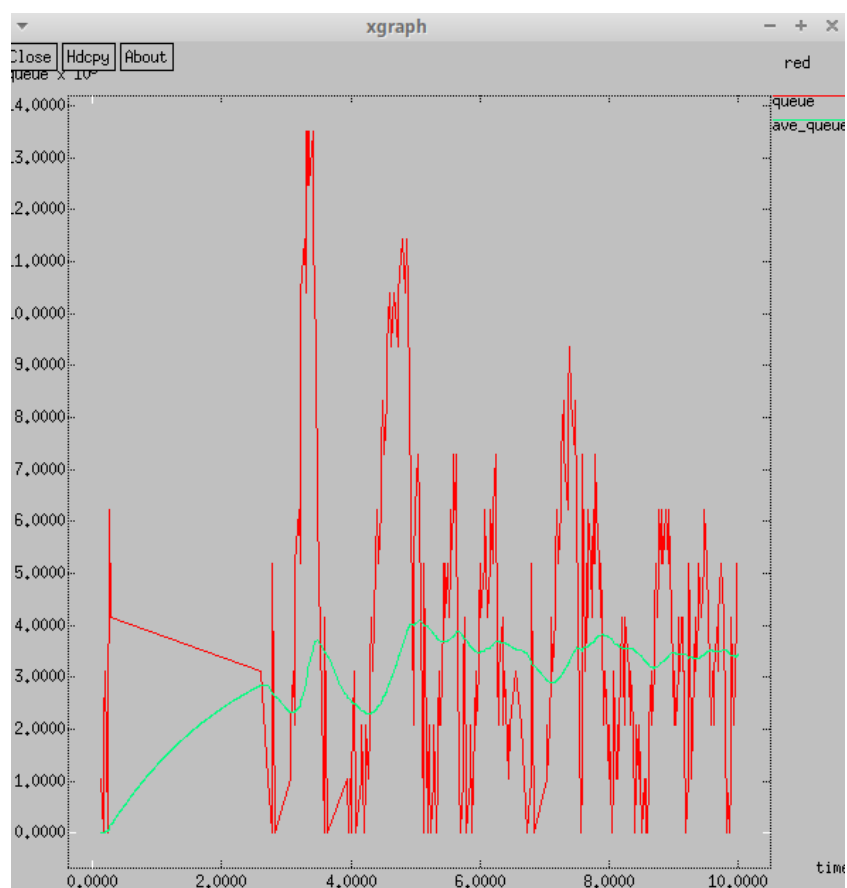


Рис. 4.3: 2

(рис. 4.4).

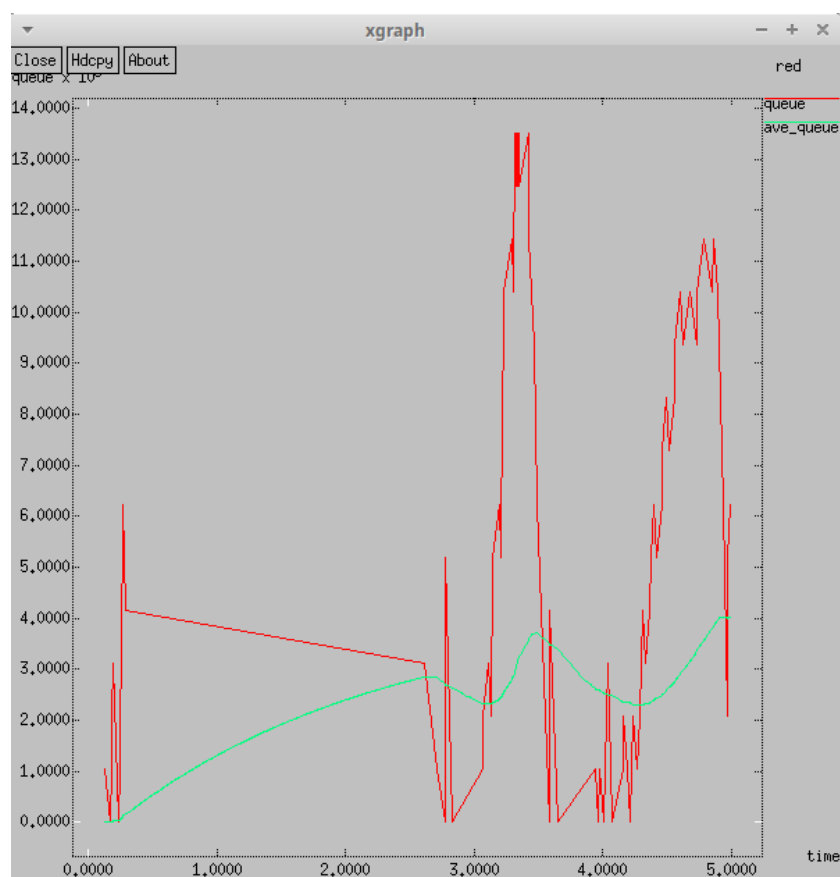


Рис. 4.4: 4

5 Выводы

Мы рассмотрели задачу Исследование протокола ТСР и алгоритма управления очередью RED, познакомится работа с протокол ТСР И алгоритма управления очередью RED .

...