Отчёт по лабораторной работе 2

Исследование протокола TCP и алгоритма управления очередью RED

Наталья Андреевна Сидорова

Содержание

Список иллюстраций

Список таблиц

1 Выполнение лабораторной работы

Я построила пример с дисциплиной RED. Сеть состоит из 6 узлов, между всеми узлами установлено дуплексное соединение с различной пропускной способностью. Узел r1 использует очередь с дисциплиной RED максимальный размер очереди 25 (рис. 1.1).

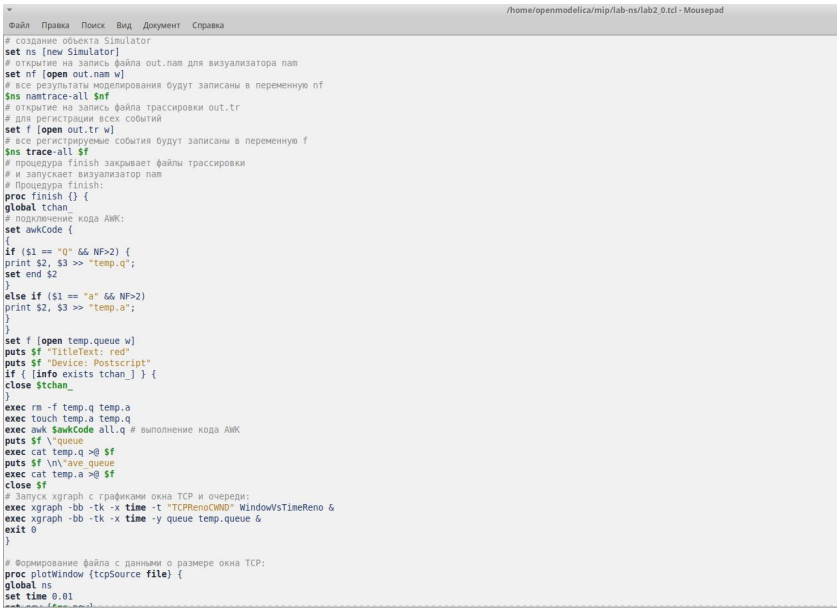

Рис. 1.1: 1 часть кода

За основу взяла код шаблона с 1 лабораторной (рис. 1.2).

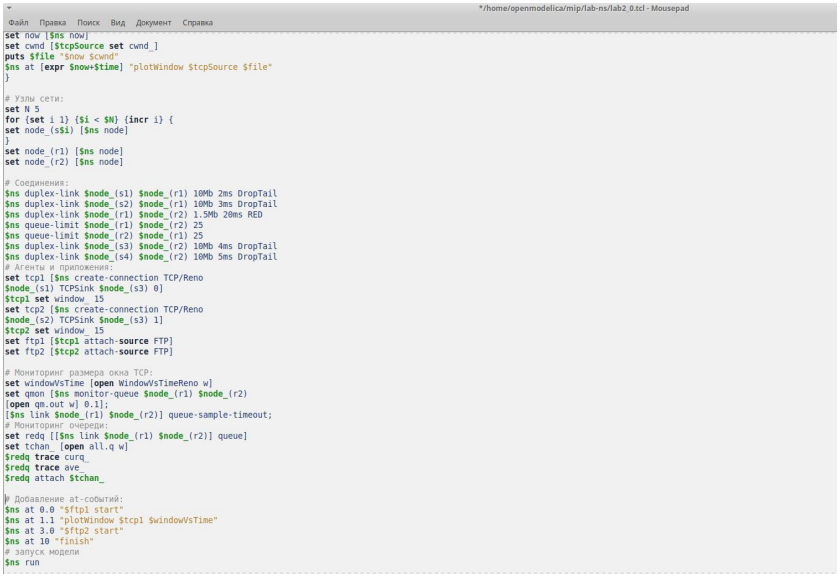

Рис. 1.2: 2 часть кода

Запустила Xgraph и посмотрела 2 графика : график изменения размера окна и график изменения размера очереди (рис. 1.3).

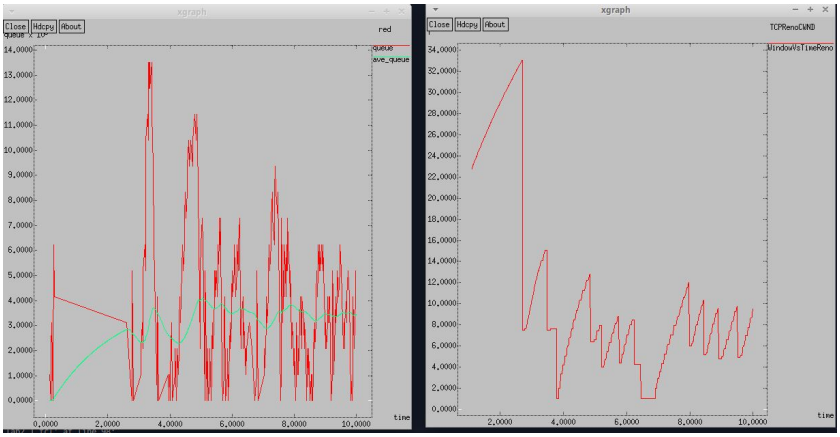

Рис. 1.3: Графики

Для выполнения упражнения я изменила на узле s1 тип протокола TCP с Reno на NewReno (рис. 1.4).

```
# 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. 1.4: NewReno

Получившийся график почти идентичен начальному (рис. 1.5).

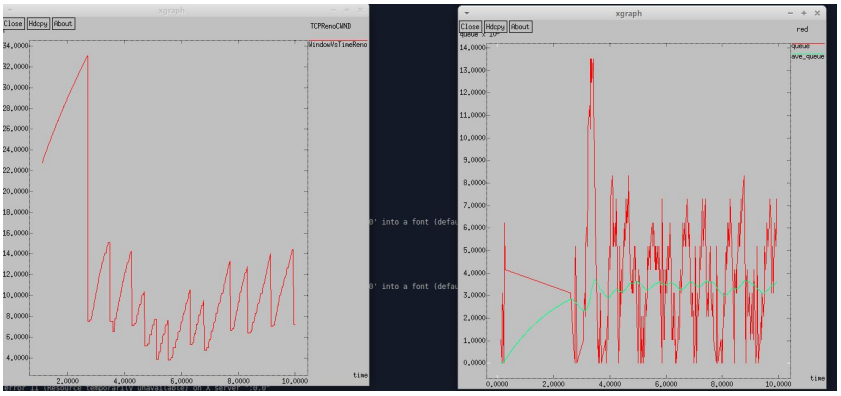

Рис. 1.5: NewReno график

Для выполнения упражнения я изменила на узле s1 тип протокола TCP с NewReno на Vegas (рис. 1.6).

```
# 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. 1.6: Vegas

Получившийся график изменения размеров окна отличается, максимальный размер окна снизился с 30+ до 20, максимальный размер очереди также снизился (рис. 1.7).

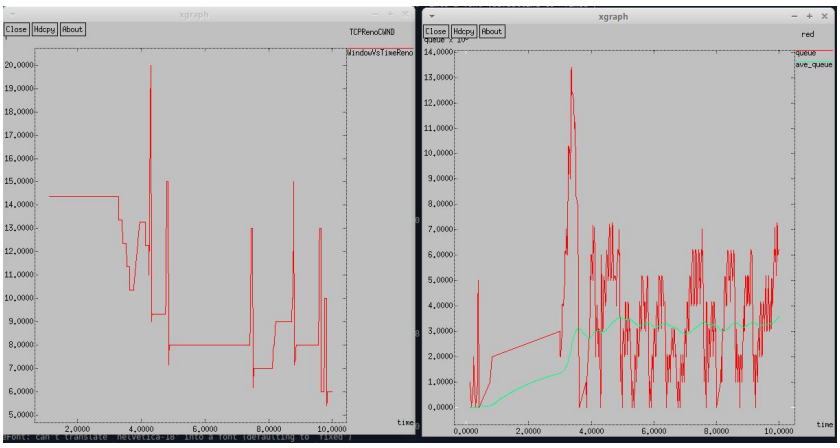

Рис. 1.7: Vegas график

Изменила цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде (рис. 1.8).

```
# Процедура finish:
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"
  puts $f "0.Color: White"
  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 \"Line"
  exec cat temp.q >@ $f
  puts $f \n\"M_Line
  exec cat temp.a >@ $f
  close $f
  # Запуск xgraph с графиками окна TCP и очереди:
  exec xgraph -fg pink -bg purple -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno &
  exec xgraph -fg green -bg purple -bb -tk -x time -y line temp.queue &
  exit 0
}
```

Рис. 1.8: Код изменения

(рис. 1.9).

```
set ftp2 [$tcp2 attach-source FTP]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "0.Color: White"
puts $windowVsTime \"Razmer_Okna"
```

Рис. 1.9: 2 часть

получившийся график (рис. 1.10).

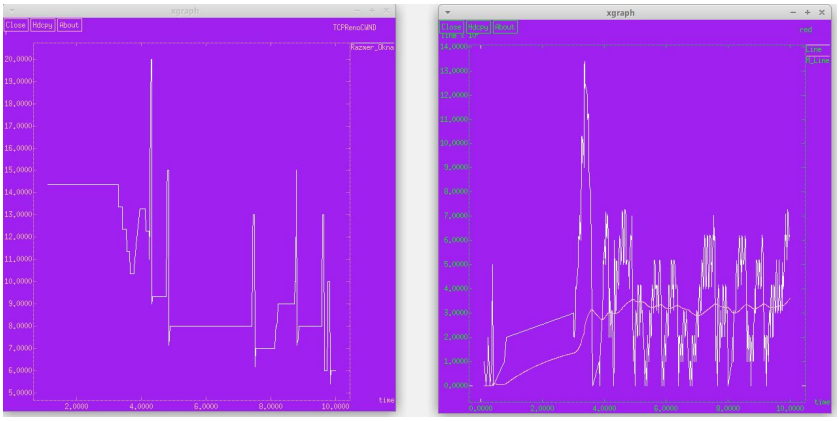

Рис. 1.10: Новый график

2 Выводы

TCP Vegas обнаруживает перегрузку в сети до того, как случайно теряется пакет, и мгновенно уменьшается размер окна. TCP Vegas обрабатывает перегрузку без потерь пакета.

Список литературы