

Лабораторная работа №2

Исследование протокола TCP и алгоритма управления очередью RED

Астраханцева А. А.

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	16

Список иллюстраций

3.1	Создание необходимых файлов и директорий	6
3.2	Создание узлов и связей	7
3.3	Создание агентов и приложений	7
3.4	Настройки размера окна и очереди	8
3.5	Процедура для файла о размере окна	8
3.6	График изменения размера окна для TCP/Reno	9
3.7	График изменения размера очереди и средней очереди для TCP/Reno	10
3.8	График изменения размера окна для TCP/Newreno	12
3.9	График изменения размера очереди и средней очереди для TCP/Newreno	13
3.10	График изменения размера окна для TCP/Vegas	14
3.11	График изменения размера очереди и средней очереди для TCP/Vegas	15

1 Цель работы

Приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования. Ознакомление с протоколом TCP и алгоритмом управления очередью RED.

2 Задание

1. Выполнение примера описания топологии сети с дисциплиной RED.
2. Выполнение упражнения

3 Выполнение лабораторной работы

1. Выполнение примера описания топологии сети с дисциплиной RED.

Постановка задачи Описание моделируемой сети: - сеть состоит из 6 узлов; - между всеми узлами установлено дуплексное соединение с различными пропускной способностью и задержкой 10 мс; - узел r1 использует очередь с дисциплиной RED для накопления пакетов, максимальный размер которой составляет 25; - TCP-источники на узлах s1 и s2 подключаются к TCP-приёмнику на узле s3; - генераторы трафика FTP прикреплены к TCP-агентам.

Сначала создадим директорию для файлов второй ЛР. Скопируем шаблон, который мы написали в первой ЛР (рис. 3.1).

```
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$ mkdir ../lab01 ../lab02
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$ mv * ../lab01/
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$ mv ../lab01 .
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$ mv ../lab02 .
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$ ls lab01
example1.tcl example3.tcl out.nam shablon.tcl
example2.tcl example4.tcl out.tr
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$ ls ..
labs-ns
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$ cp lab01/shablon.tcl lab02
openmodelica@openmodelica-VirtualBox:~/mip/labs-ns$
```

Рис. 3.1: Создание необходимых файлов и директорий

Начнем с создания узлов. Далее зададим необходимые параметры соединений между узлами (рис. 3.2).

```

# Узлы сети:
set N 5
for {set i 1} {$i < $N} {incr i} {
    set node_(s$i) [$ns node]
}
set node_(r1) [$ns node]
set node_(r2) [$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

```

Рис. 3.2: Создание узлов и связей

Далее необходимо по условиям задачи создать tcp-агентов и поверх наложить ftp-протокол (рис. 3.3).

```

# 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]

```

Рис. 3.3: Создание агентов и приложений

Зададим мониторинг размера окна TCP и мониторинг очереди (рис. 3.4).

```
# 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_
```

Рис. 3.4: Настройки размера окна и очереди

Далее необходимо описать процедуру для формирования файла с данными о размере окна TCP. Потом добавляем at-события (рис. 3.5).

```
# Формирование файла с данными о размере окна TCP:
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"

# Добавление at-событий:
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"
```

Рис. 3.5: Процедура для файла о размере окна

Запуская написанный скрипт, мы увидим 2 открывшихся окна xgraph. Первое окно отображает то, как менялся размер окна (поле Размер окна (Window) (длина 16 бит) содержит количество байт, которое может быть послано после байта, получение которого уже подтверждено). (рис. 3.6).

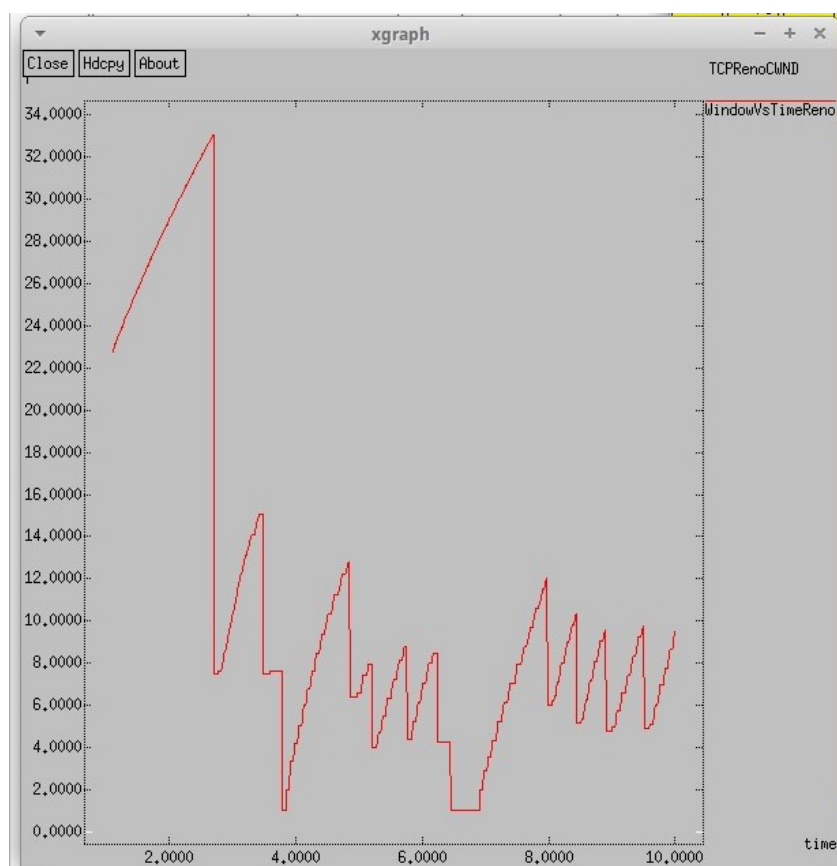


Рис. 3.6: График изменения размера окна для TCP/Reno

Вторым окном откроется график изменения размера очереди и средней очереди (рис. 3.7).

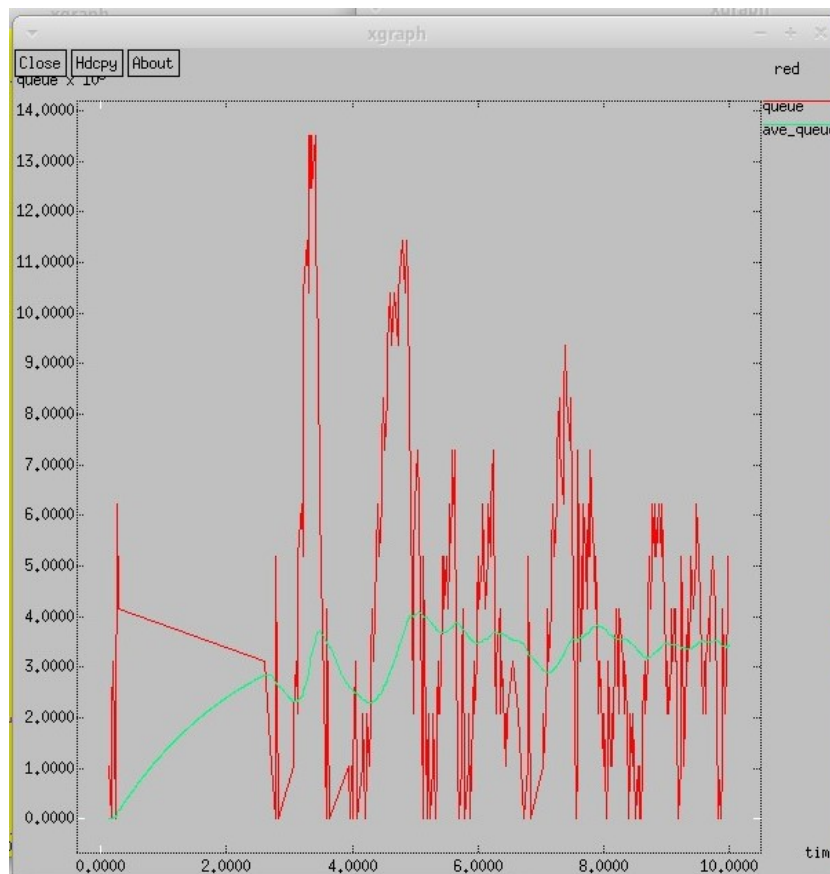


Рис. 3.7: График изменения размера очереди и средней очереди для TCP/Reno

2. Выполнение упражнения

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

Внесем изменения в имеющийся скрипт. Для начала изменим тип TCP с Reno на Newreno для первого узла.

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]
```

В процедуре finish для графика изменения длины очереди и средней длины очереди изменяем цвет графика.

```
set f [open temp.queue w]
puts $f "TitleText: red"
puts $f "Device: Postscript"
puts $f "0.Color: Purple"
puts $f "1.Color: Green"
```

Изменяем подписи в легенде для графика изменения длины очереди и средней длины очереди изменяем цвет осей.

```
puts $f \"Current_size_of_queue
exec cat temp.q >@ $f
puts $f \"Average_size_of_queue
exec cat temp.a >@ $f
```

Изменяем подписи в легенде и цвет графика для изменения размера окна.

```
puts $windowVsTime "0.Color: Purple"
puts $windowVsTime \"Window_size"
```

Снова открываются 2 окна. График изменения размера окна (рис. 3.8).

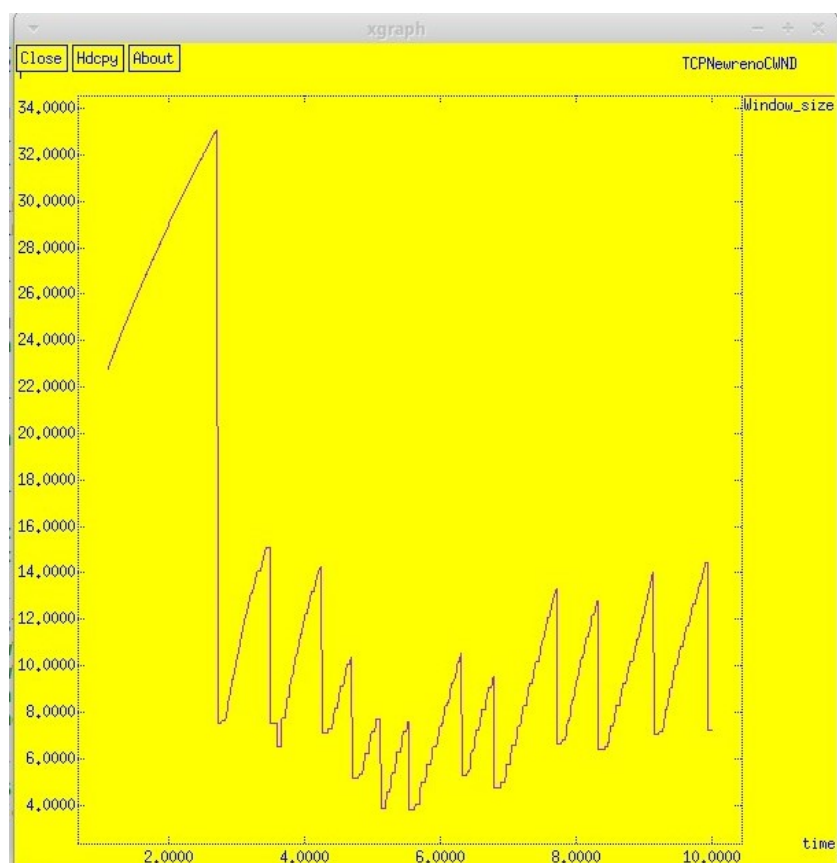


Рис. 3.8: График изменения размера окна для TCP/Newreno

Вторым окном откроется график изменения размера очереди и средней очереди (рис. 3.9).

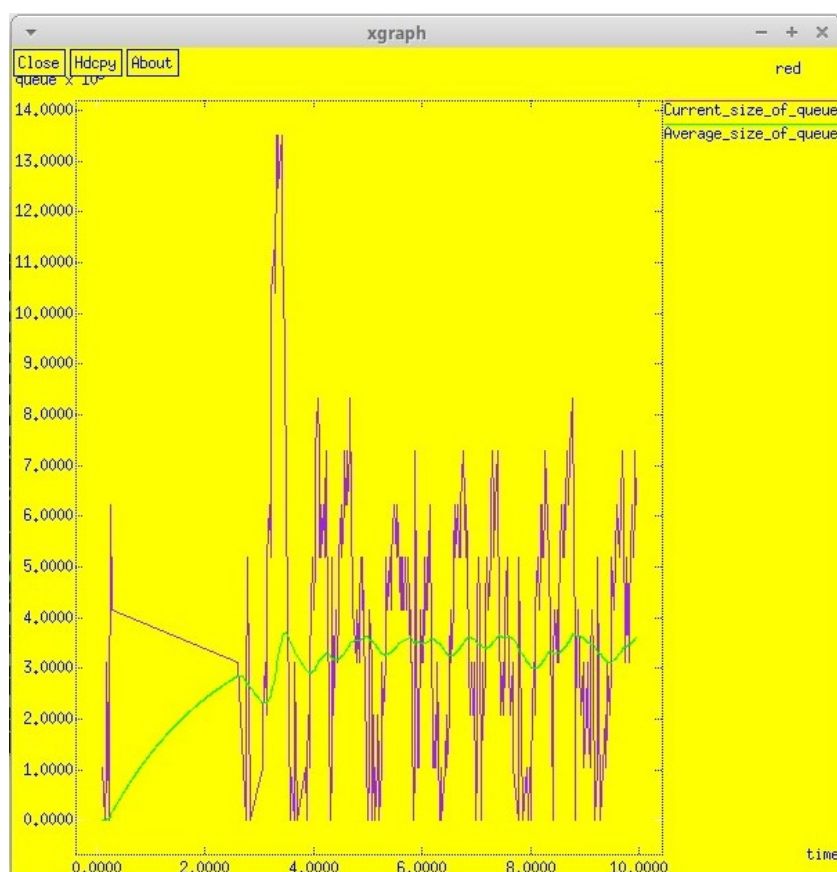


Рис. 3.9: График изменения размера очереди и средней очереди для TCP/Newreno

После этого внесем изменения в имеющийся скрипт: изменим тип TCP с Newreno на Vegas для первого узла.

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]
```

Снова открываются 2 окна. График изменения размера окна (рис. 3.10).

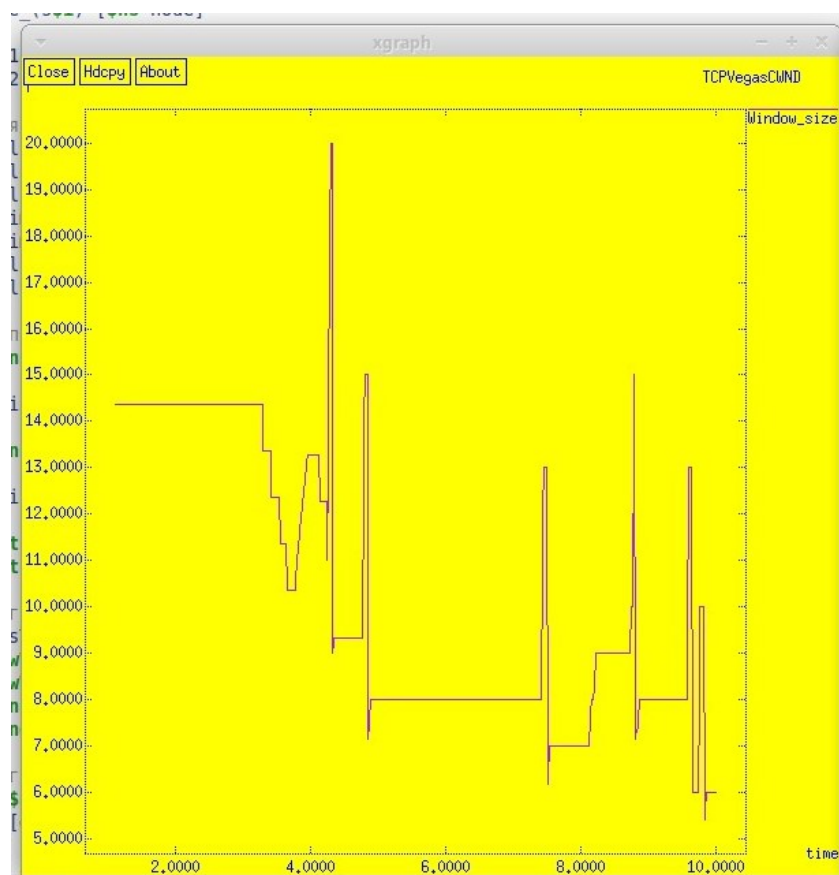


Рис. 3.10: График изменения размера окна для TCP/Vegas

Вторым окном откроется график изменения размера очереди и средней очереди (рис. 3.11).



Рис. 3.11: График изменения размера очереди и средней очереди для TCP/Vegas

4 Выводы

В ходе выполнения лабораторной работы я приобрела навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также провела анализ полученных результатов моделирования. Помимо этого, я ознакомилась с протоколом TCP и алгоритмом управления очередью RED.