

Лабораторная работа №2

Исследование протокола TCP и алгоритма управления очередью RED

Астраханцева А. А.

20 февраля 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Астраханцева Анастасия Александровна
- НФИбд-01-22, 1132226437
- Российский университет дружбы народов
- 1132226437@pfur.ru
- <https://github.com/aaastrakhantseva>



Вводная часть

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.

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

Выполнение ЛР

- 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.

Создание необходимых директорий и файла

```
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. 1: Создание необходимых файлов и директорий

```
# 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. 2: Создание узлов и связей

```
# 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: Создание агентов и приложений

```
# 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. 4: Настройки размера окна и очереди

```
# 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. 5: Процедура для файла о размере окна

График размера окна Reno

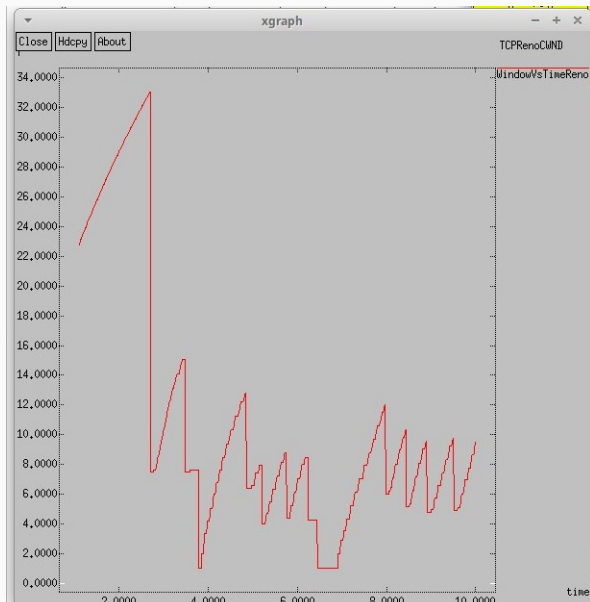
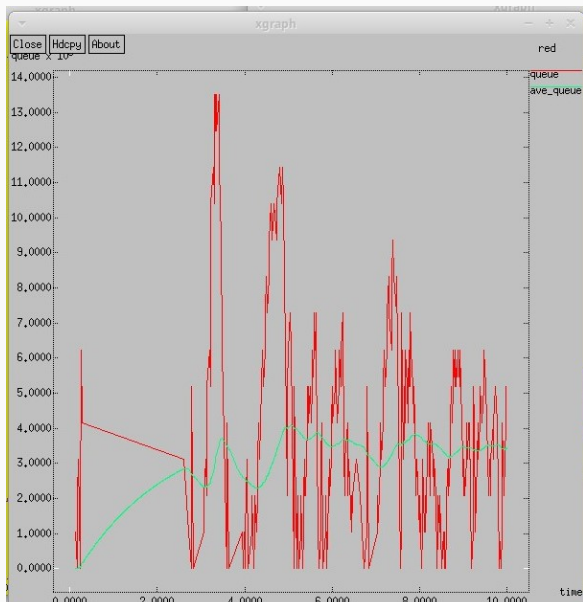


График изменения размера очереди и средней очереди Reno



Выполнение упражнения

- 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).

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]
```

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

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

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

График размера окна Newreno

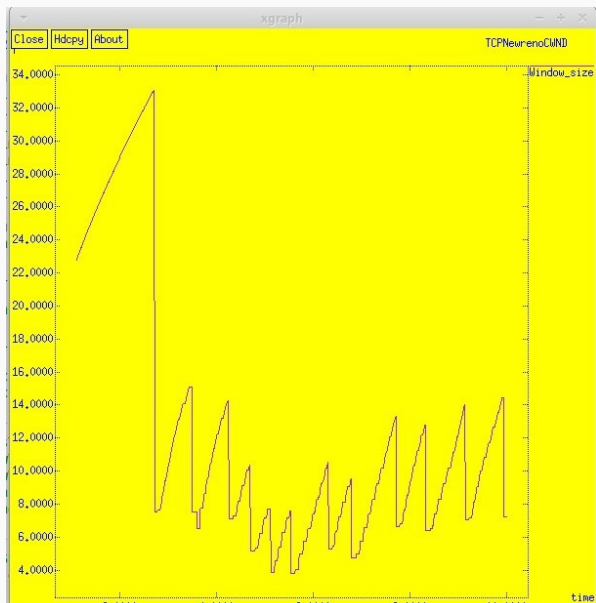
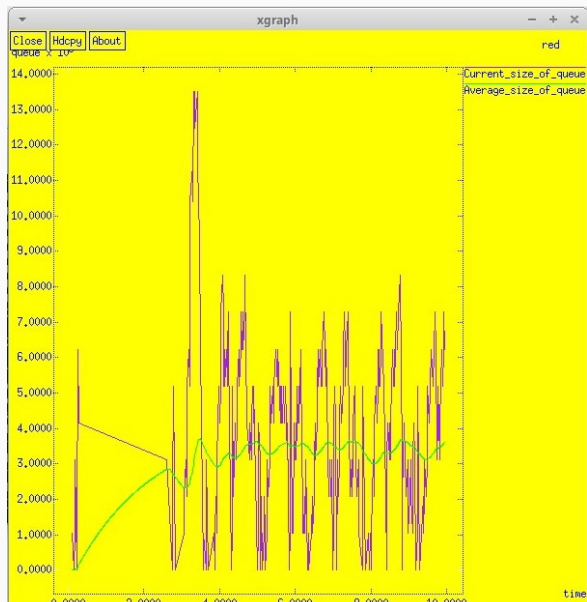


График изменения размера очереди и средней очереди Newreno



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]
```


График размера окна Vegas

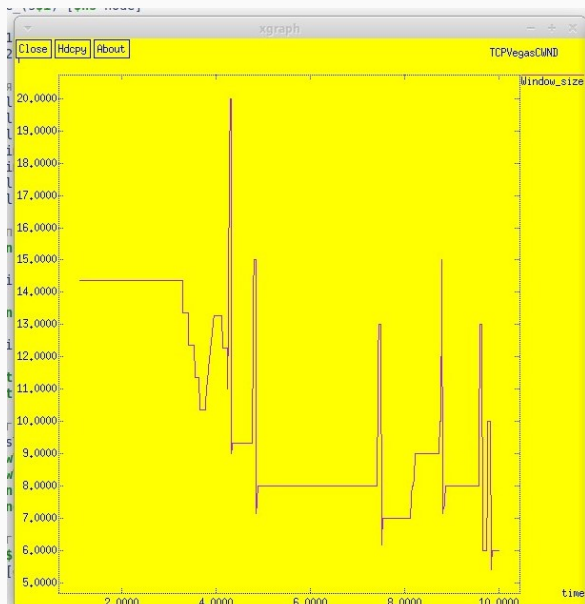


График изменения размера очереди и средней очереди Vegas



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.

Спасибо за внимание!
