Отчёт по лабораторной работе №2

Имитационное моделирование

Екатерина Канева, НФИбд-02-22

Содержание

# 1 Цель работы

Научиться работать с мониторингом очередей, TCP, дисциплиной RED, сравнить различные протоколы.

# 2 Задание

На основе имеющейся модели изменить тип протокола с Reno на NewReno, а затем на Vegas, сравнить. Также, внести изменения при отображении окон с графиками.

# 3 Теоретическое введение

Протокол управления передачей (Transmission Control Protocol, TCP) имеет средства управления потоком и коррекции ошибок, ориентирован на установление соединения.

# 4 Выполнение лабораторной работы

Для начала я создала файл с уже описанным в тексте кодом, запустила, чтобы сделать модель, отвечающую следующим требованиям:

* 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.

Получились следующие графики (рис. [[1](#fig:1)] и [[2](#fig:2)]):

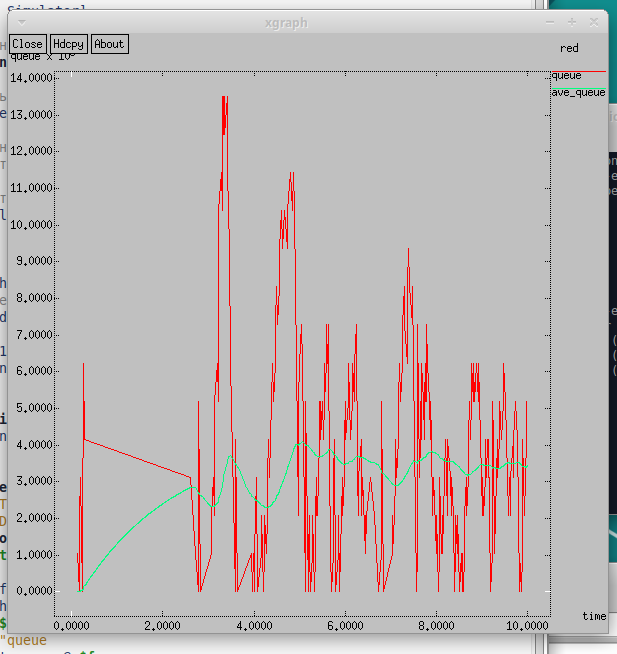


Figure 1: Изменение длины очереди.

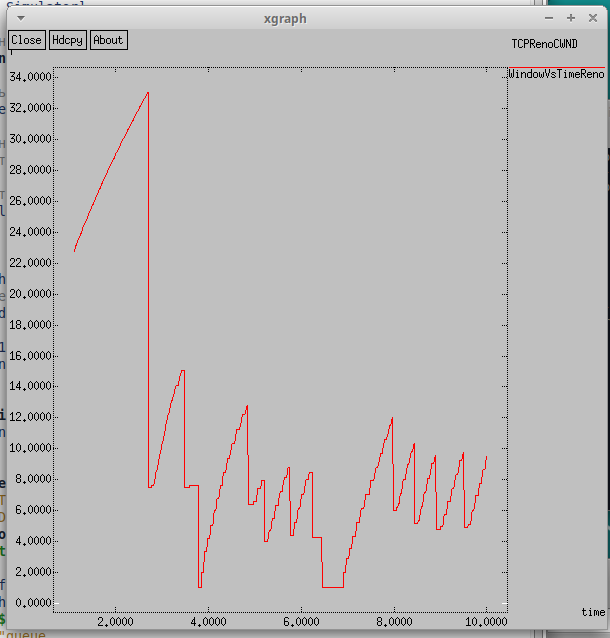


Figure 2: Изменение TCP-окна.

Далее, следуя заданию, изменим Reno для s1 на NewReno. Для этого изменила следующую часть кода:

set tcp1 [$ns create-connection TCP/Newreno $node\_(s1) TCPSink $node\_(s3) 0]

Получились следующие графики (рис. [[3](#fig:3)] и [[4](#fig:4)]):

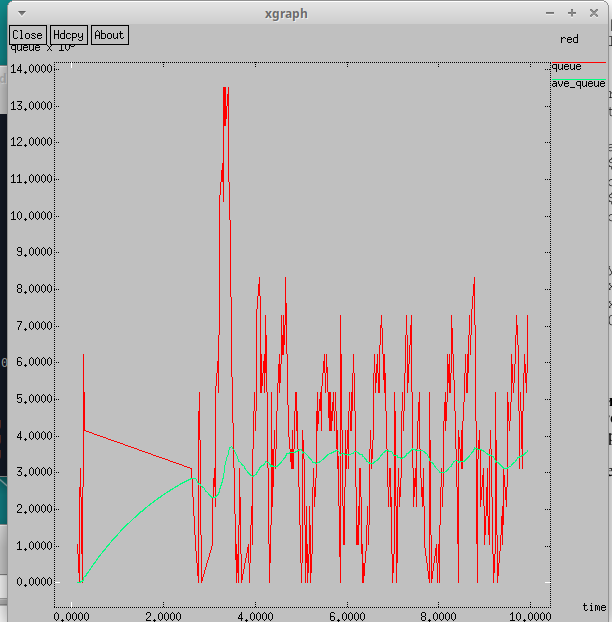


Figure 3: Изменение длины очереди.

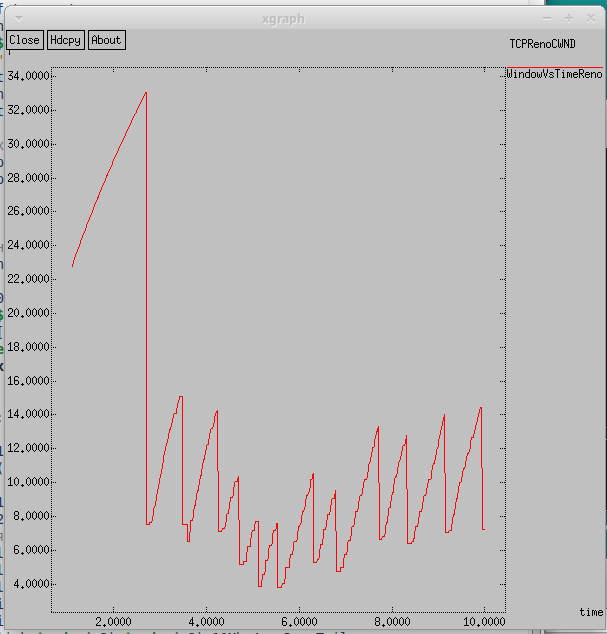


Figure 4: Изменение TCP-окна.

Далее, следуя заданию, изменим Newreno для s1 на Vegas. Для этого изменила следующую часть кода:

set tcp1 [$ns create-connection TCP/Vegas $node\_(s1) TCPSink $node\_(s3) 0]

Получились следующие графики (рис. [[5](#fig:5)] и [[6](#fig:6)]):

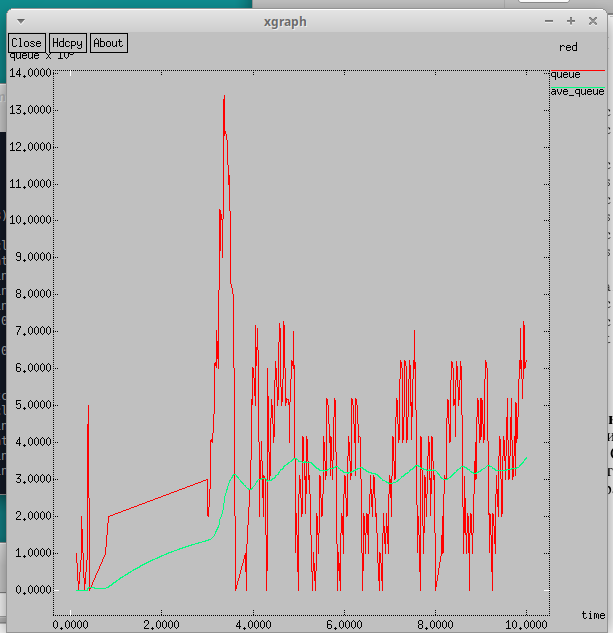


Figure 5: Изменение длины очереди.

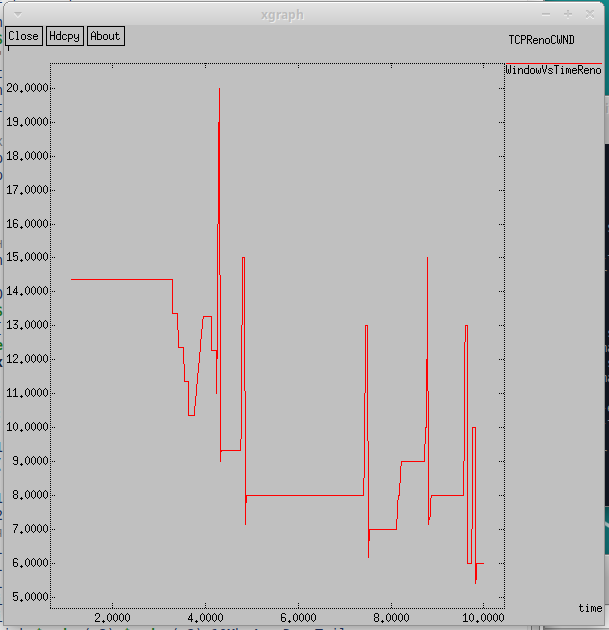


Figure 6: Изменение TCP-окна.

По графикам видно, что средняя длина очереди почти не поменялась, она так и лежит в пределах 3. Однако на графике изменения окна видно, что максимальное значение у Vegas значительно меньше, а значит Vegas лучше обрабатывает большую нагрузку и теряет меньше пакетов.

Далее я внесла изменения при отображении окон с графиками (измените цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде). Для этого внесла изменения в процедуру finish:

puts $f "TitleText: red"  
 puts $f "Device: Postscript"  
 puts $f "0.Color: Blue"  
 puts $f "1.Color: Green"  
 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 \"Ochered"  
 exec cat temp.q >@ $f  
 puts $f \n\"Sr\_Ochered"  
 exec cat temp.a >@ $f  
 close $f  
 # Запуск xgraph с графиками окна TCP и очереди:  
 exec xgraph -fg red -bg pink -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno &  
 exec xgraph -fg white -bg pink -bb -tk -x time -y ochered temp.queue &  
 exit 0

Получились следующие графики (рис. [[7](#fig:7)] и [[8](#fig:8)]):



Figure 7: Изменение длины очереди.

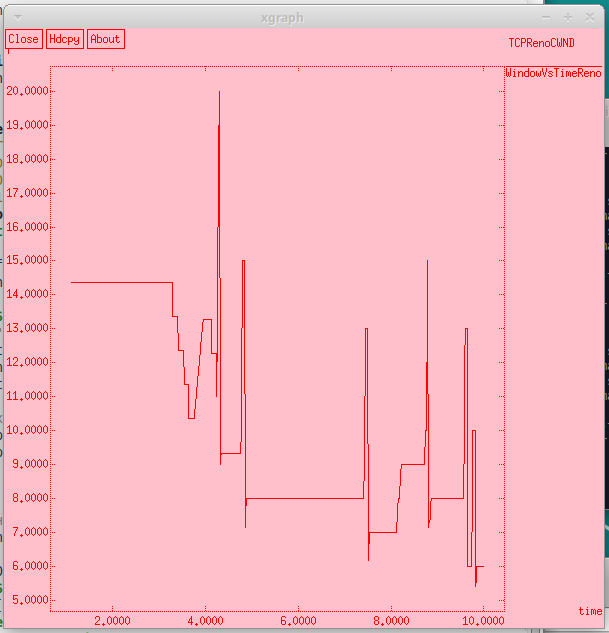


Figure 8: Изменение TCP-окна.

# 5 Выводы

Научилась работать с мониторингом очередей, TCP, дисциплиной RED, сравнила различные протоколы.