

Лабораторная работа 2

Имитационное моделирование

Голощапов Ярослав Вячеславович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	11

Список иллюстраций

4.1	Реализация модели	8
4.2	Графики модели	9
4.3	код для NewRepo	9
4.4	Измененные графики	10
4.5	Сравнение графиков	10

Список таблиц

1 Цель работы

Исследование протокола TCP и алгоритма управления очередью RED

2 Задание

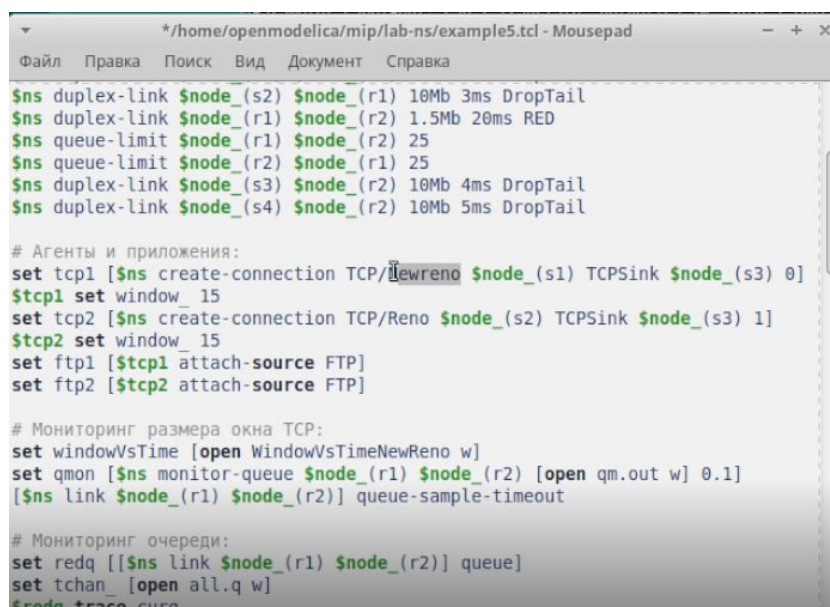
Получить практические навыки на примерах и выполнить упражнение

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

Постановка задачи Описание моделируемой сети: – сеть состоит из 6 узлов; – между всеми узлами установлено дуплексное соединение с различными пропускной способностью и задержкой 10 мс (см. рис. 2.4); – 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. На рис. 2.4 приведена схема моделируемой сети. Требуется разработать сценарий, реализующий модель согласно рис. 2.4, построить в Xgraph график изменения TCP-окна, график изменения длины очереди и средней длины очереди. (рис. 4.1). (рис. 4.2).



```
#!/home/openmodelica/mip/lab-ns/example5.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

$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

# 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]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeNewReno 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
```

Рис. 4.1: Реализация модели



Рис. 4.2: Графики модели

Упражнение – 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). (рис. 4.3) (рис. 4.4).

```

* /home/openmodelica/mip/lab-ns/example5.tcl - Mousepad
Файл Правка Поиск Вид Документ Справка

# 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_
Здесь curq_ – текущий размер очереди, ave_ – средний размер очереди.
# Добавление at-событий:
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"
# Формирование файла с данными о размере окна TCP:
proc plotWindow {tcpSource file} {
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

```

Рис. 4.3: код для NewReno

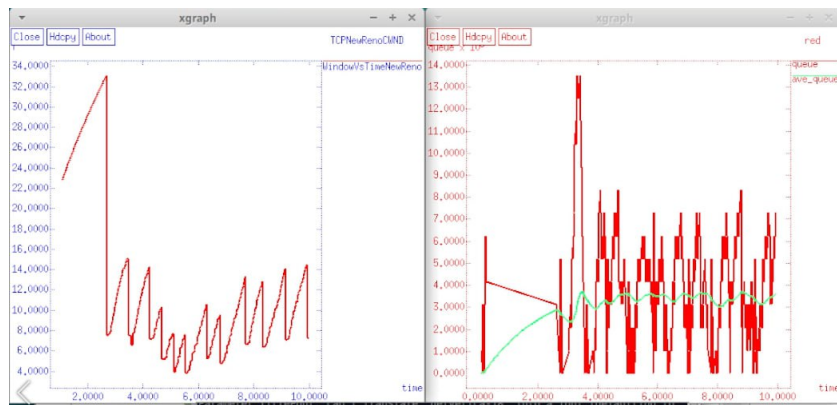


Рис. 4.4: Измененные графики

Модель Vegas (рис. 4.5)

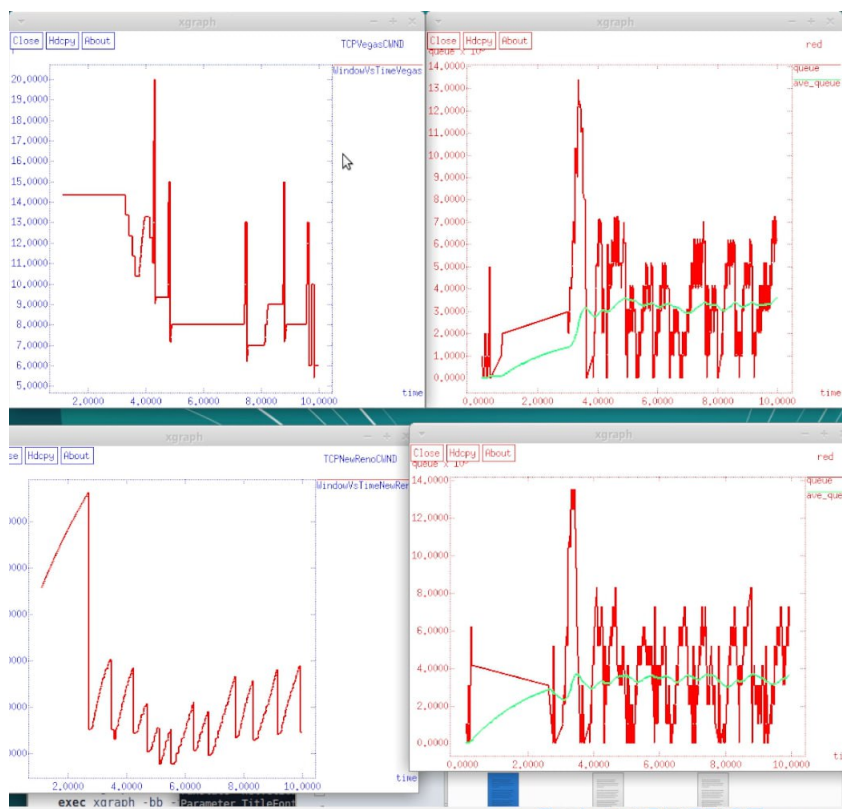


Рис. 4.5: Сравнение графиков

5 Выводы

В этой лабораторной работе я исследовал протокол ТСР и алгоритм управления очередью RED