

# Лабораторная работа №2

Исследование протокола TCP и алгоритма управления очередью RED

---

Акопян Сатеник

Российский университет дружбы народов, Москва, Россия

Исследование протокола TCP и алгоритма управления очередью RED.

Описание моделируемой сети:

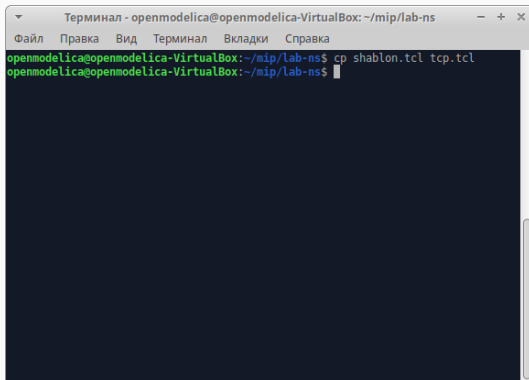
- сеть состоит из 6 узлов;
- между всеми узлами установлено дуплексное соединение с различными пропускной способностью и задержкой 10 мс;
- узел r1 использует очередь с дисциплиной RED для накопления пакетов, максимальный размер которой составляет 25;
- TCP-источники на узлах s1 и s2 подключаются к TCP-приёмнику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

Требуется разработать сценарий, реализующий модель согласно, построить в Xgraph график изменения TCP-окна, график изменения длины очереди и средней длины очереди

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

Объект мониторинга очереди оповещает диспетчера очереди о поступлении пакета. Диспетчер очереди осуществляет мониторинг очереди.

1. Создается файл tcp.tcl на основе шаблона shablon.tcl с реализацией модели



The image shows a terminal window titled "Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns". The window has a menu bar with "Файл", "Правка", "Вид", "Терминал", "Вкладки", and "Справка". The terminal content shows the prompt "openmodelica@openmodelica-VirtualBox:~/mip/lab-ns\$" followed by the command "cp shablon.tcl tcp.tcl" and the prompt "openmodelica@openmodelica-VirtualBox:~/mip/lab-ns\$" with a cursor.

```
Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp shablon.tcl tcp.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$
```

Рис. 1: рисунок 1

## Выполнение лабораторной работы

[illegible]

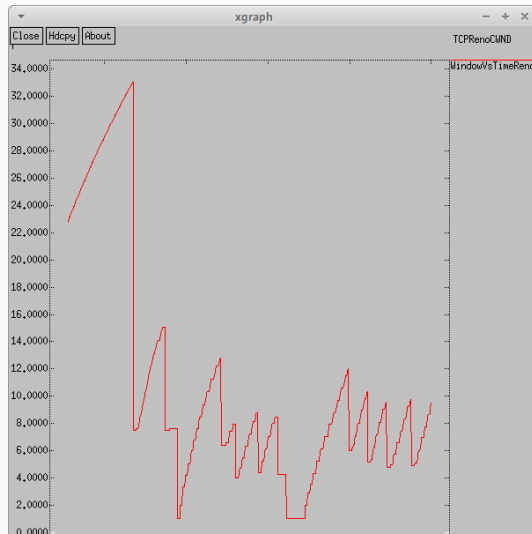
Рис. 2: рисунок 2

## Выполнение лабораторной работы

[illegible]

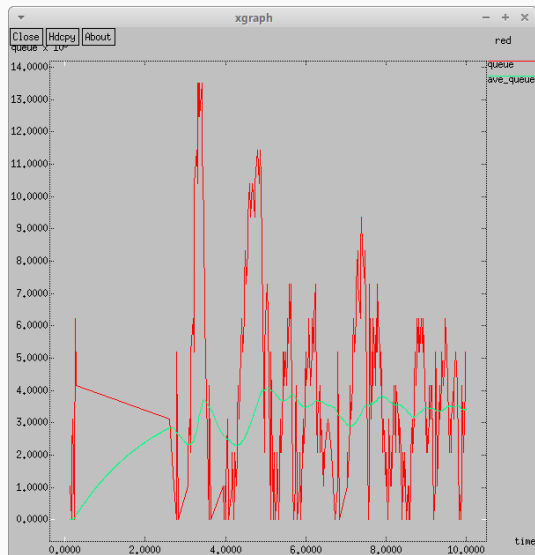
Рис. 3: рисунок 3

## Графики окна TCP и очереди





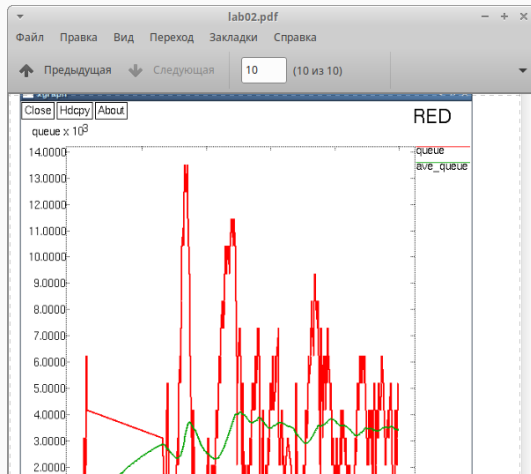
# Выполнение лабораторной работы

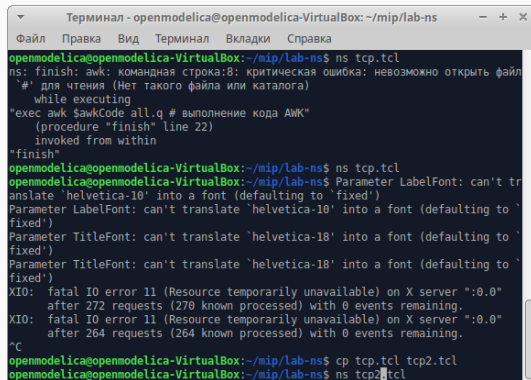


## Выполнение лабораторной работы

2. Измените в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas.

Меняем TCP/Reno -> TCP/Newreno

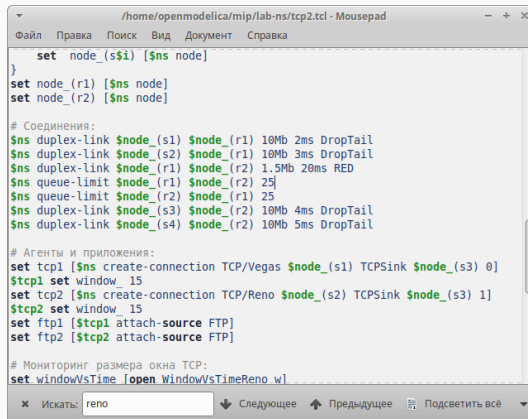




```
Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns tcp.tcl
ns: finish: awk: командная строка:8: критическая ошибка: невозможно открыть файл
'#' для чтения (Нет такого файла или каталога)
while executing
"exec awk $awkCode all.q # выполнение кода AWK"
(procedure "finish" line 22)
invoked from within
"finish"
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns tcp.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ Parameter LabelFont: can't tr
anslate 'helvetica-10' into a font (defaulting to 'fixed')
Parameter LabelFont: can't translate 'helvetica-10' into a font (defaulting to `
fixed')
Parameter TitleFont: can't translate 'helvetica-18' into a font (defaulting to `
fixed')
Parameter TitleFont: can't translate 'helvetica-18' into a font (defaulting to `
fixed')
XIO: fatal IO error 11 (Resource temporarily unavailable) on X server ":0.0"
after 272 requests (270 known processed) with 0 events remaining.
XIO: fatal IO error 11 (Resource temporarily unavailable) on X server ":0.0"
after 264 requests (264 known processed) with 0 events remaining.
^C
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp tcp.tcl tcp2.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns tcp2.tcl
```

Рис. 7: рисунок 7

Меняем TCP/Reno -> TCP/Vegas



```

/home/openmodelica/mip/lab-ns/tcp2.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

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

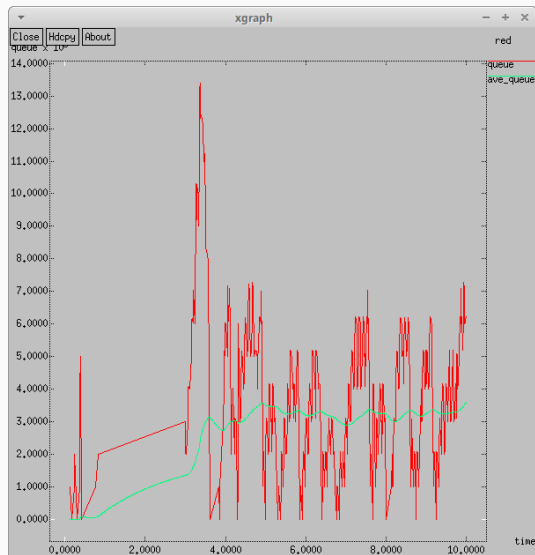
# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Vegas $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeReno w]
```

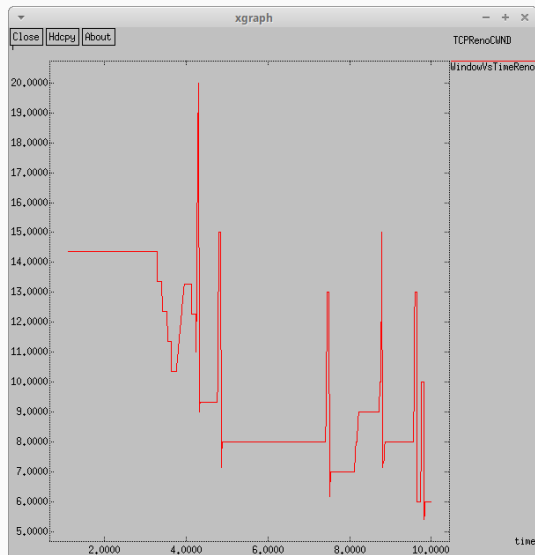
Искать: reno    ↓ Следующее    ↑ Предыдущее    Подсветить всё

Рис. 8: рисунок 8

# Выполнение лабораторной работы



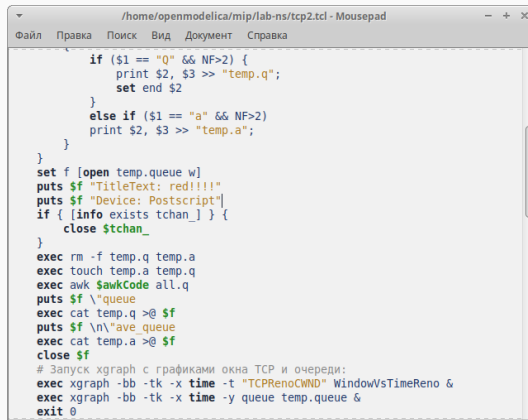
# Выполнение лабораторной работы



## Выполнение лабораторной работы

3. Внесите изменения при отображении окон с графиками (измените цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде)

Поменяла red -> red!!!!



```
/home/openmodelica/mip/lab-ns/tcp2.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

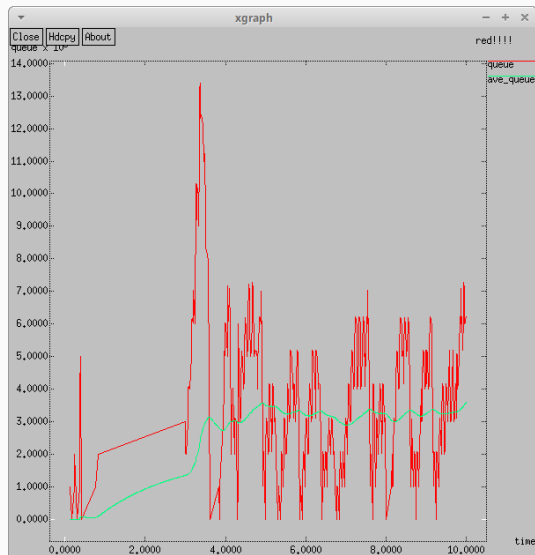
if ($1 == "Q" && NF>2) {
    print $2, $3 >> "temp.q";
    set end $2
}
else if ($1 == "a" && NF>2)
    print $2, $3 >> "temp.a";
}
}

set f [open temp.queue w]
puts $f "TitleText: red!!!!"
puts $f "Device: Postscript"
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 "\"queue"
exec cat temp.q >@ $f
puts $f "\\n\"ave_queue"
exec cat temp.a >@ $f
close $f

# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -bb -tk -x time -t "TCPrenoCWND" WindowVsTimeReno &
exec xgraph -bb -tk -x time -y queue temp.queue &
exit 0
```

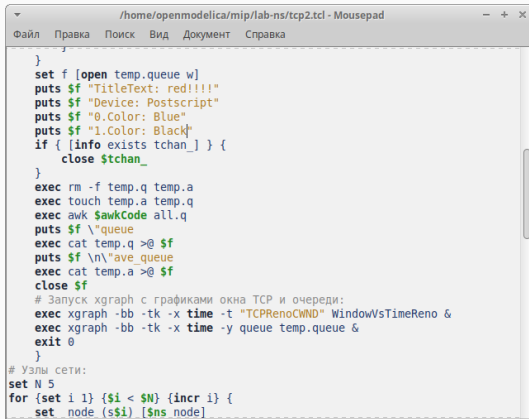
# Выполнение лабораторной работы





# Выполнение лабораторной работы

Поменяла цвета траекторий:



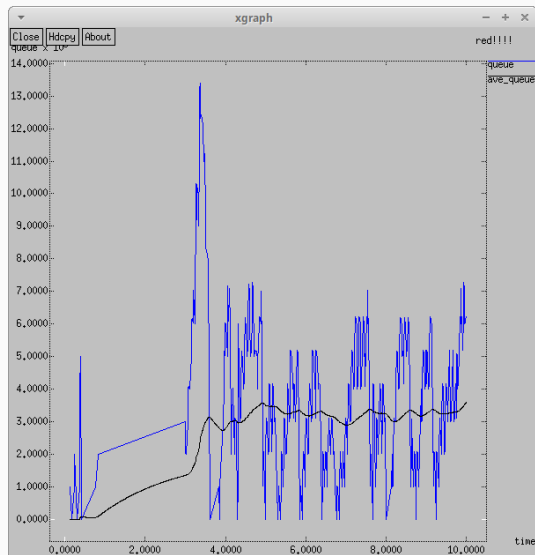
```
/home/openmodelica/mip/lab-ns/tcp2.tcl - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

}
set f [open temp.queue w]
puts $f "TitleText: red!!!"
puts $f "Device: Postscript"
puts $f "0.Color: Blue"
puts $f "1.Color: Black"
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 "\"queue"
exec cat temp.q >@ $f
puts $f "\n\"ave_queue"
exec cat temp.a >@ $f
close $f
# Запуск xgraph с графиками окна TCP и очереди:
exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno &
exec xgraph -bb -tk -x time -y queue temp.queue &
exit 0
}

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

Рис. 13: рисунок 13

# Выполнение лабораторной работы



В результате данной лабораторной работы я исследовала протокол TCP и алгоритм управления очередью RED.