

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

дисциплина: Моделирование информационных процессов

Студент:

Ибрагимов Улугбек

Группа:

НФИбд-02-20

МОСКВА

2023 г.

|  |    |
|--|----|
| Цель .....   | 3  |
| Задачи .....   | 4  |
| Выполнение лабораторной работы.....                            | 5  |
| Реализация модели .....  | 5  |
| Кастомизация внешнего вида графика .....                       | 9  |
| Изменения в модели на узле s1 тип протокола TCP на NewReno     | 10 |
| Изменения в модели на узле s1 тип протокола TCP на Vegas ..... | 12 |
| Анализ результатов .....                                       | 13 |
| Вывод.....   | 14 |

## **Цель**

Приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

## Задачи

1. Реализовать модель данную в примере.
2. Изменить в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas.
3. Внести изменения при отображении окон с графиками (измените цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде).

# Выполнение лабораторной работы

## Реализация модели

### Постановка задачи:

- сеть состоит из 6 узлов;
- между всеми узлами установлено дуплексное соединение с различными пропускной способностью и задержкой 10 мс (см. рис. 1);
- 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.

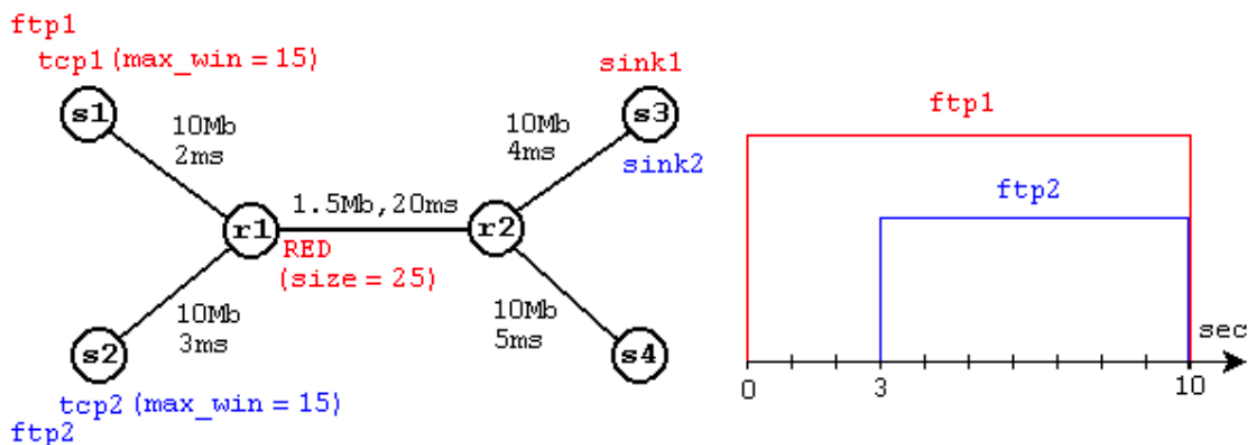


Рис. 1. Схема сети

Требуется разработать сценарий, реализующий модель согласно рис. 1, построить в Xgraph график изменения TCP-окна, график изменения длины очереди и средней длины очереди.

1. Возьмем за основу, написанный шаблон. Перед строкой [`$ns at 5.0 «finish»`] будем писать модель. Сначала опишем сеть:

```
# 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

# 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]

```

2. Добавим мониторинг размера окна TCP и мониторинг очереди. `window_` — верхняя граница окна приёмника (Advertisment Window) TCP соединения:

```

# 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_

```

3. Добавим процедуру 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"
}

```

4. Изменим процедуру finish. Здесь cwnd\_ — текущее значение окна перегрузки:

```
proc finish {} {
    global tchan_
    # подключение кода AWK:
    set awkCode {
        {
            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
    # выполнение кода AWK
    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 "TCPRegoCWND" WindowVs-
TimeReno &
    exec xgraph -bb -tk -x time -y queue temp.queue &
    exit 0
}
```

5. Добавим at-события и изменим строку с at-событием finish. Заметим, что здесь

curq\_ — текущий размер очереди, ave\_ — средний размер очереди.

```
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"
```

6. Запустим симуляцию (рис. 2-3):

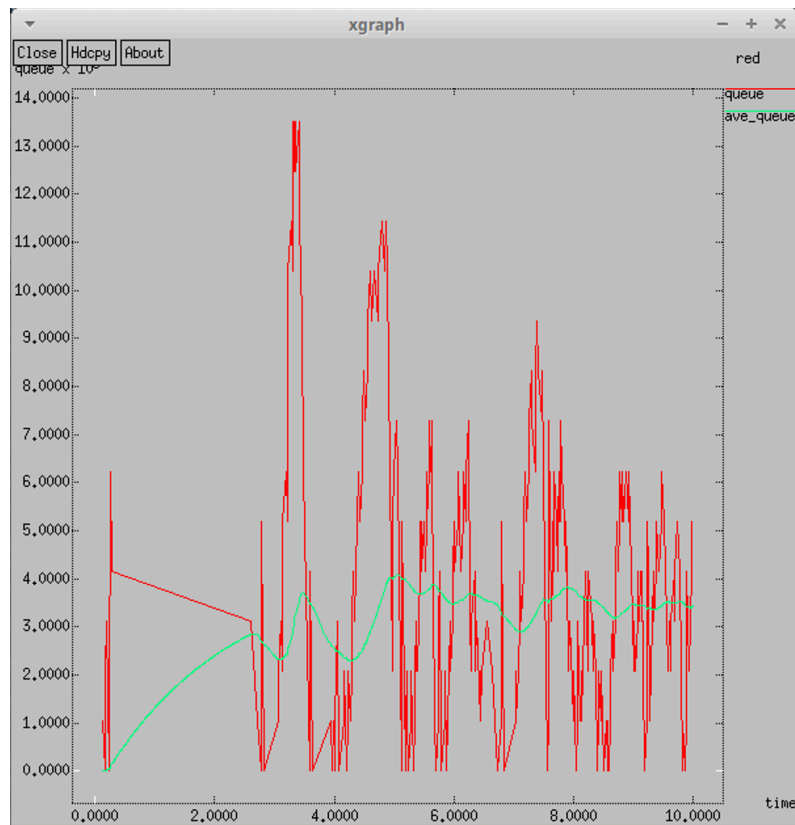


Рис. 2. График динамики длины очереди и средней длины очереди (Reno)

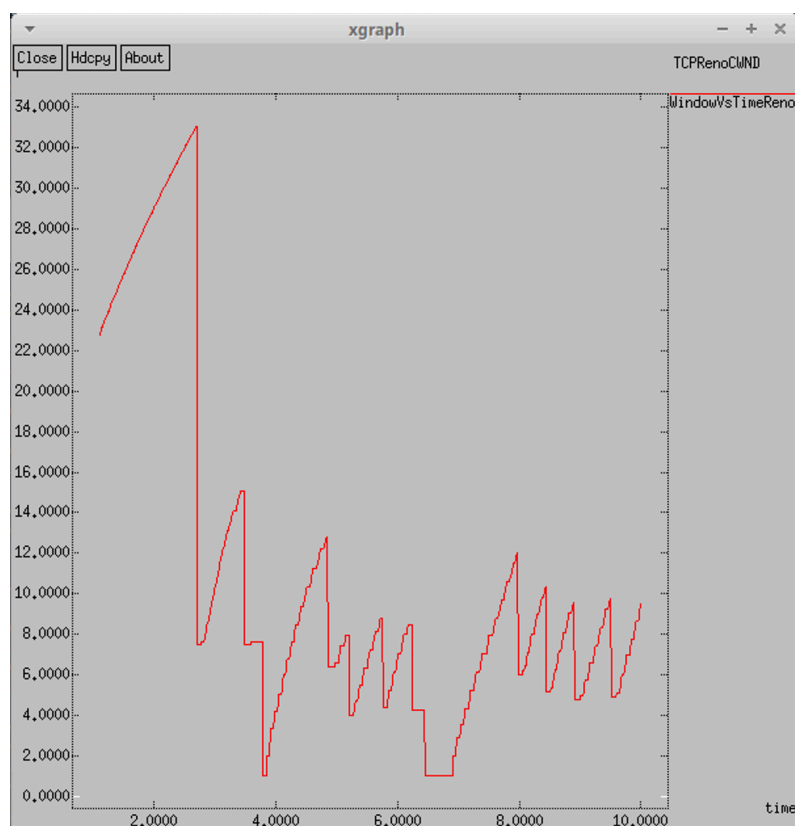


Рис. 3. График динамики размера окна TCP (Reno)



## *Кастомизация внешнего вида графика*

1. Изменим цвет фона: для этого перед в строках с запуском xgraph с графиками окна TCP и очереди добавим флаг 'bg' и укажем цвет. Для изменения подписей осей изменим значения после флагов 'x' и 'y'.

Было

```
exec xgraph -bb -tk -x time -t "TCPRenoCWND" WindowVsTimeReno &
exec xgraph -bb -tk -x time -y queue temp.queue &
exit 0
```

Стало

```
exec xgraph -bg white -x t -y WindowSize -bb -tk -t "TCPNew-
RenoCWND" WindowVsTimeNewReno &
exec xgraph -bg white -x t -y Queue -bb -tk temp.queue &
exit 0
```

2. Изменим цвет траекторий графика динамики длины очереди и средней длины очереди:

Было

```
exec awk $awkCode all.q
puts $f \"queue
exec cat temp.q >@ $f
puts $f \"ave_queue
exec cat temp.a >@ $f
close $f
```

Стало

```
exec awk $awkCode all.q
puts $f \"CueQueue
exec cat temp.q >@ $f
puts $f \"AveQueue
exec cat temp.a >@ $f
puts $f "0.Color: green"
puts $f "1.Color: blue"
close $f
```

3. Изменим цвет траекторий графика динамики размера окна TCP для этого добавим после предыдущего блока кода:

```
global windowVsTime
puts $windowVsTime "0.Color: green"
```

4. Добавим подпись траекторий в легенде графика динамики длины очереди и средней длины очереди:

Было

```
exec awk $awkCode all.q
puts $f \"queue
exec cat temp.q >@ $f
puts $f \n\"ave_queue
exec cat temp.a >@ $f
```

Стало

```
exec awk $awkCode all.q
puts $f \"CueQueue
exec cat temp.q >@ $f
puts $f \n\"AvgQueue
exec cat temp.a >@ $f
```

5. Для изменения подписи в троекторий в легенде графика динамики размера окна

можно менять названия файла для записи:

```
exec xgraph -bg white -x t -y WindowSize -bb -tk -t "TCPNewReno-
CWND" WindowVsTimeNewReno &
set windowVsTime [open WindowVsTimeNewReno w]
```

### ***Изменения в модели на узле s1 тип протокола TCP на NewReno***

1. Изменим тип протокола TCP на узле s1 на NewReno:

```
set tcp1 [$ns create-connection TCP/Newreno $node_(s1)
TCPSink $node_(s3) 0]
```

2. Запустим симуляцию (рис. 4-5):

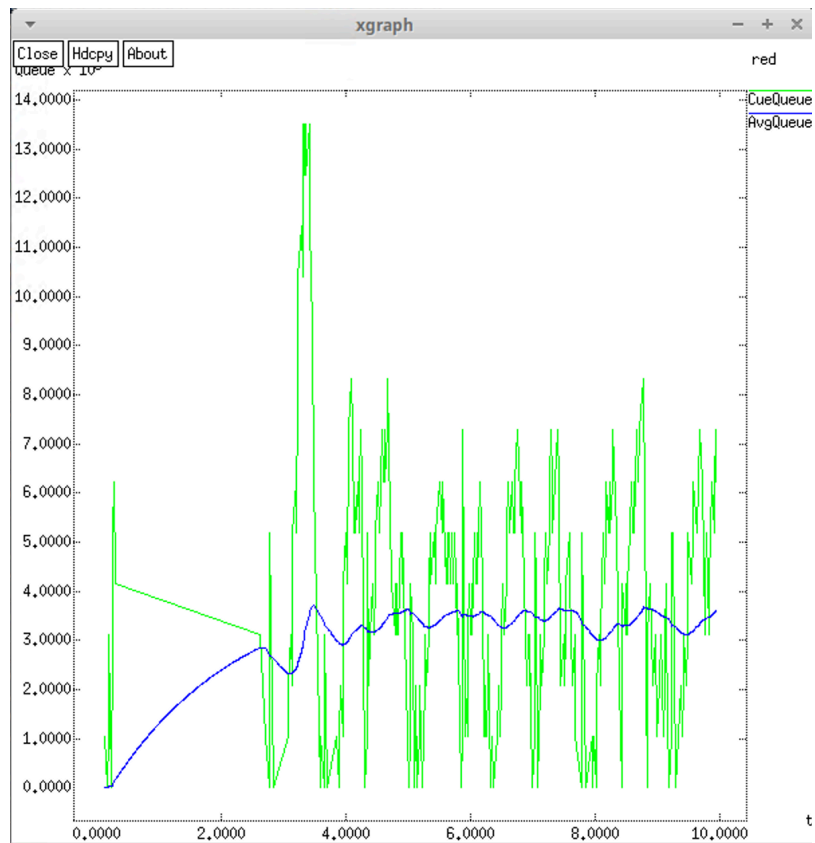


Рис. 4. График динамики длины очереди и средней длины очереди (NewReno)

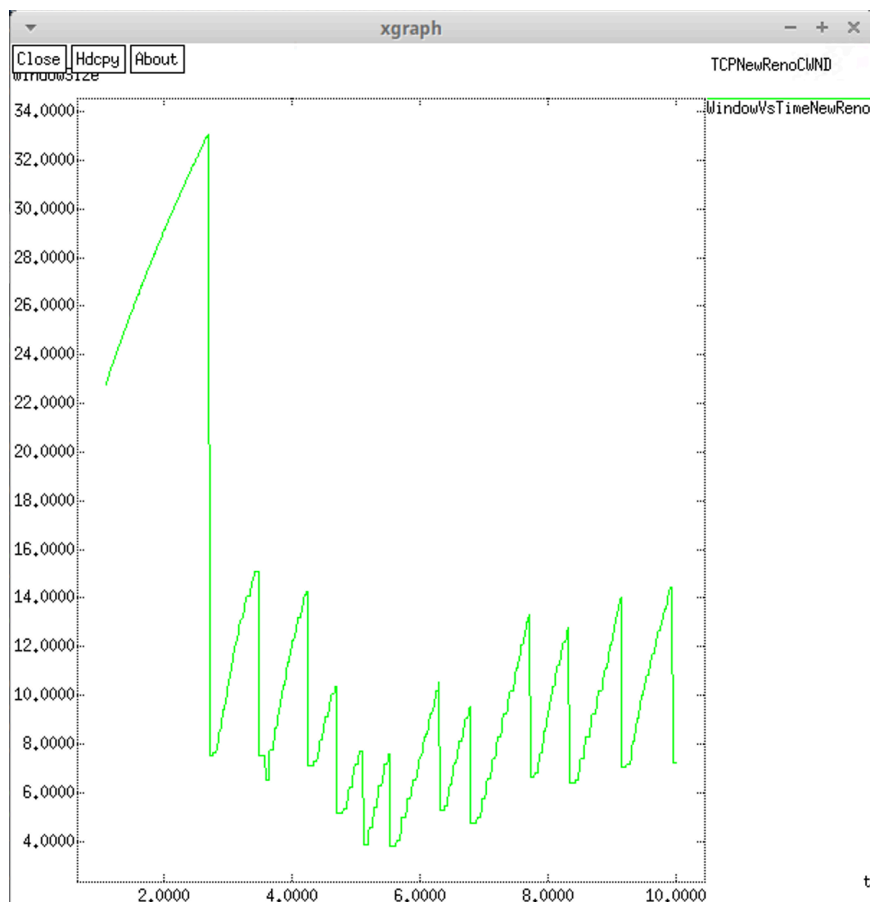


Рис. 5. График динамики размера окна TCP (NewReno)

## *Изменения в модели на узле s1 тип протокола TCP на Vegas*

1. Изменим тип протокола TCP на узле s1 на NewReno:

```
set tcp1 [$ns create-connection TCP/Vegas $node_(s1) TCPSink  
$node_(s3) 0]
```

2. Запустим симуляцию (рис. 6-7):

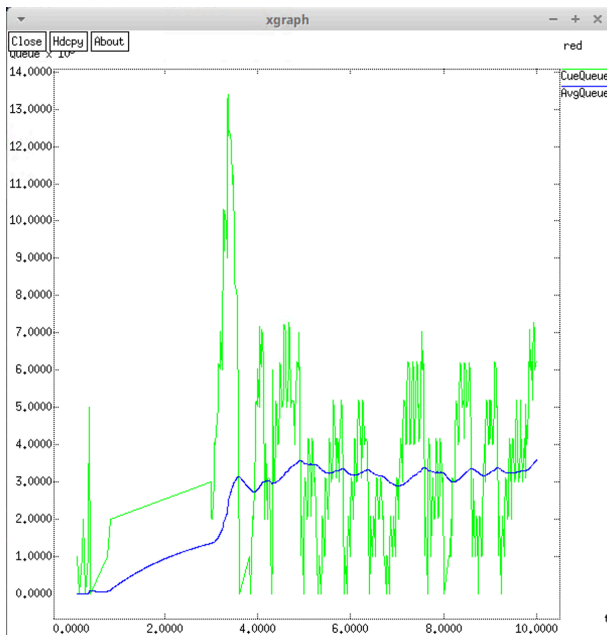


Рис. 6. График динамики длины очереди и средней длины очереди (Vegas)

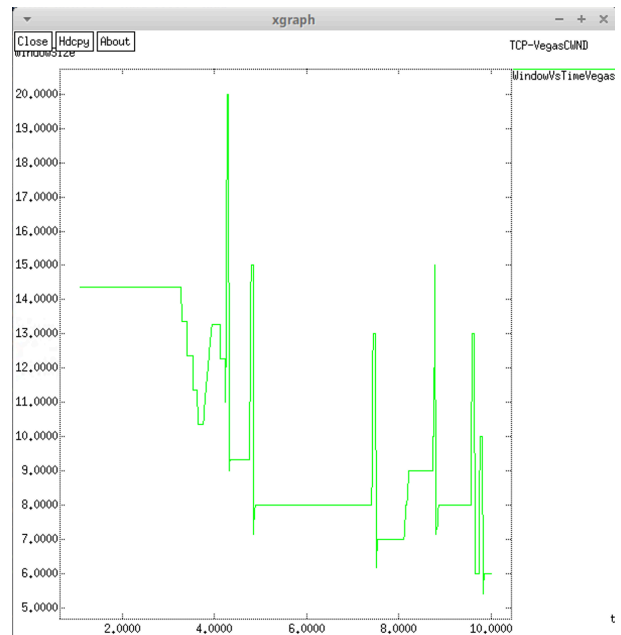


Рис. 7. График динамики размера окна TCP (Vegas)

## Анализ результатов

В рамках выполнения работы произошли некоторые незначительные проблемы, но все они решены. В остальном работа выполнена без непредвиденных проблем в соответствии с руководством: ошибок и сбоев не произошло.

Стоит отметить, что написание скриптов NS-2 происходи крайне комфортно, синтаксис, объекты интуитивно понятны. Также сами скрипты довольно быстро выполняются.

Среда xgraph также оказалась удобной для использования. Но для реализации поставленных, задачи необходимо было использовать документацию и руководство текущего курса.

## **Вывод**

В результате выполнения работы, были получены и улучшены практические навыки моделирования на NS-2. Также были созданы, настроены и смоделирована сеть. Также был произведен мониторинг очередей и увиден алгоритм red на практике. Более того, были приобретены навыки использования xgraph.