

# **Отчёт по лабораторной работе №2**

**Имитационное моделирование**

**Екатерина Канева, НФИбд-02-22**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>18</b>

## Список иллюстраций

4.1	Изменение длины очереди. . . . .	9
4.2	Изменение ТСП-окна. . . . .	10
4.3	Изменение длины очереди. . . . .	11
4.4	Изменение ТСП-окна. . . . .	12
4.5	Изменение длины очереди. . . . .	13
4.6	Изменение ТСП-окна. . . . .	14
4.7	Изменение длины очереди. . . . .	16
4.8	Изменение ТСП-окна. . . . .	17

## **Список таблиц**

# **1 Цель работы**

**Научиться работать с мониторингом очередей, ТСП, 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.

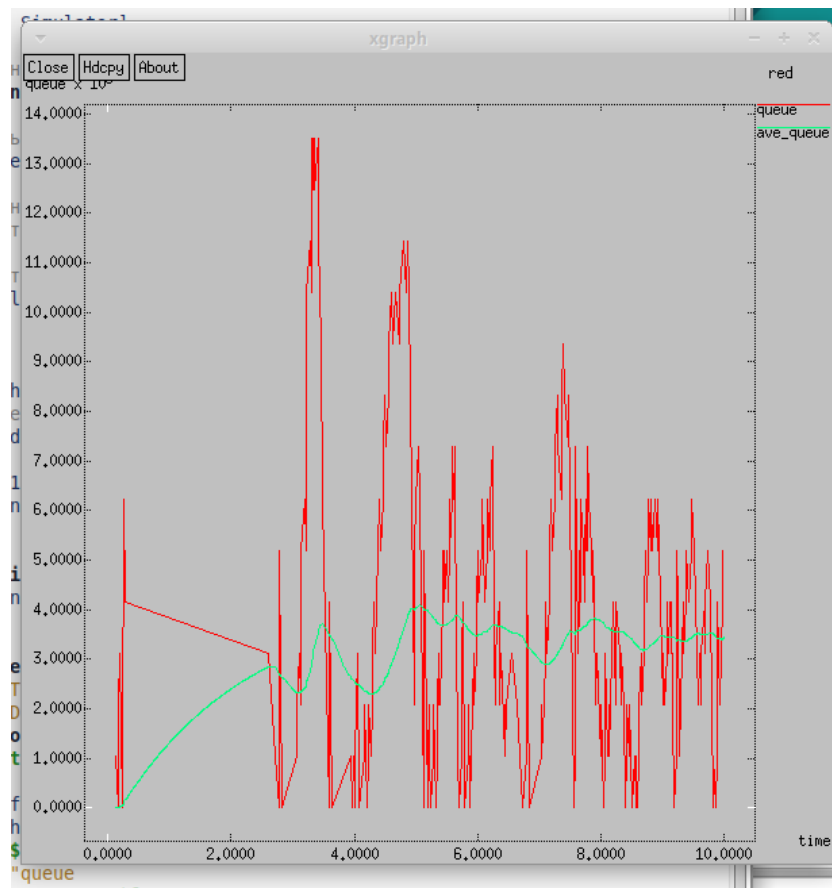
## 4 Выполнение лабораторной работы

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

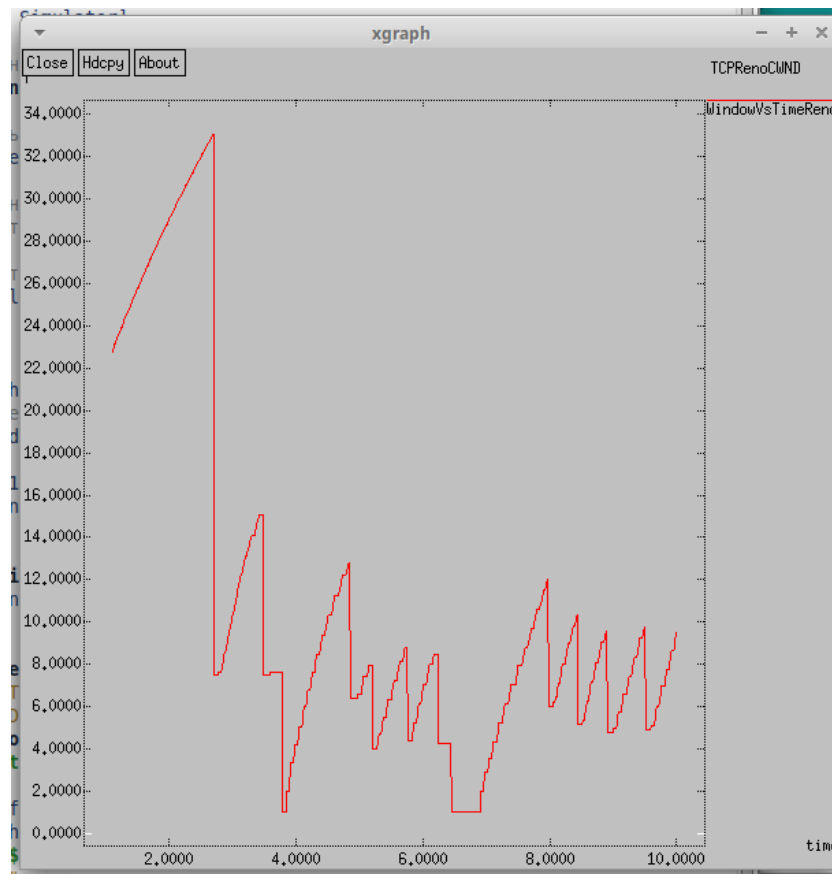
- сеть состоит из 6 узлов;
- между всеми узлами установлено дуплексное соединение с различными пропускной способностью и задержкой 10 мс;
- узел r1 использует очередь с дисциплиной RED для накопления пакетов, максимальный размер которой составляет 25;
- TCP-источники на узлах s1 и s2 подключаются к TCP-приёмнику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

Получились следующие графики (рис. [4.1] и [4.2]):





**Рис. 4.1: Изменение длины очереди.**

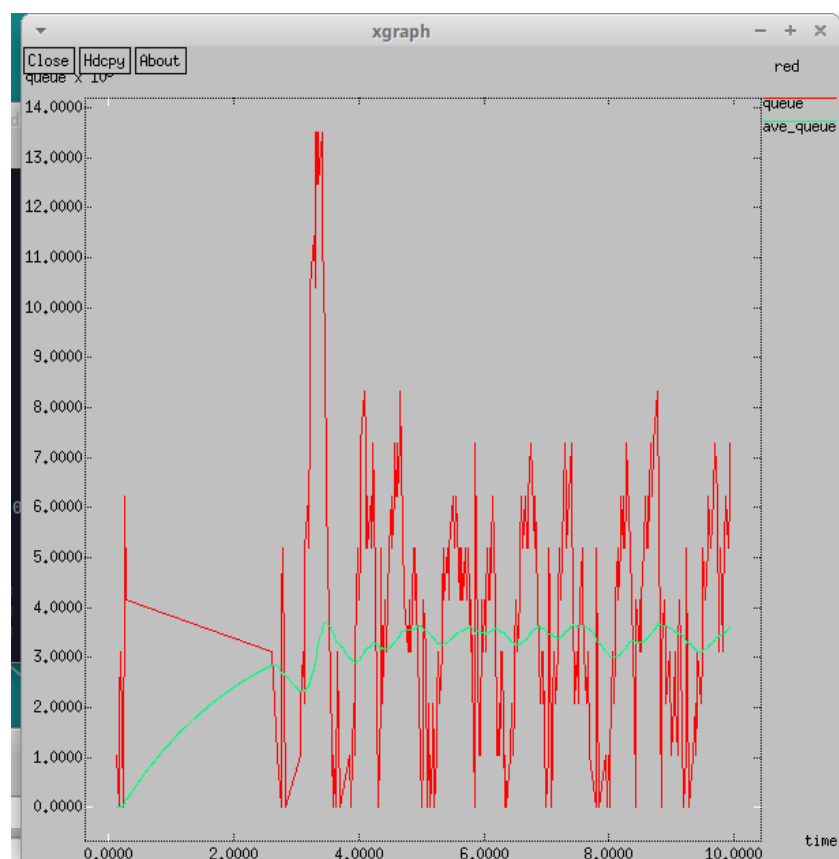


**Рис. 4.2: Изменение TCP-окна.**

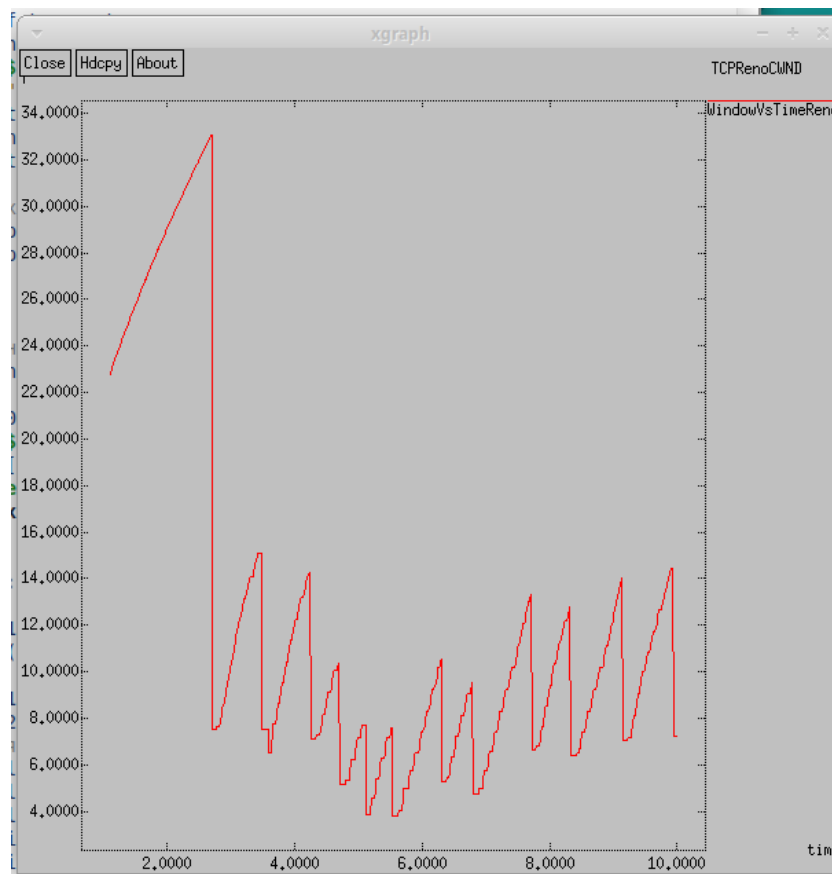
Далее, следуя заданию, изменим Reno для s1 на NewReno. Для этого изменила следующую часть кода:

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

Получились следующие графики (рис. [4.3] и [4.4]):



**Рис. 4.3: Изменение длины очереди.**

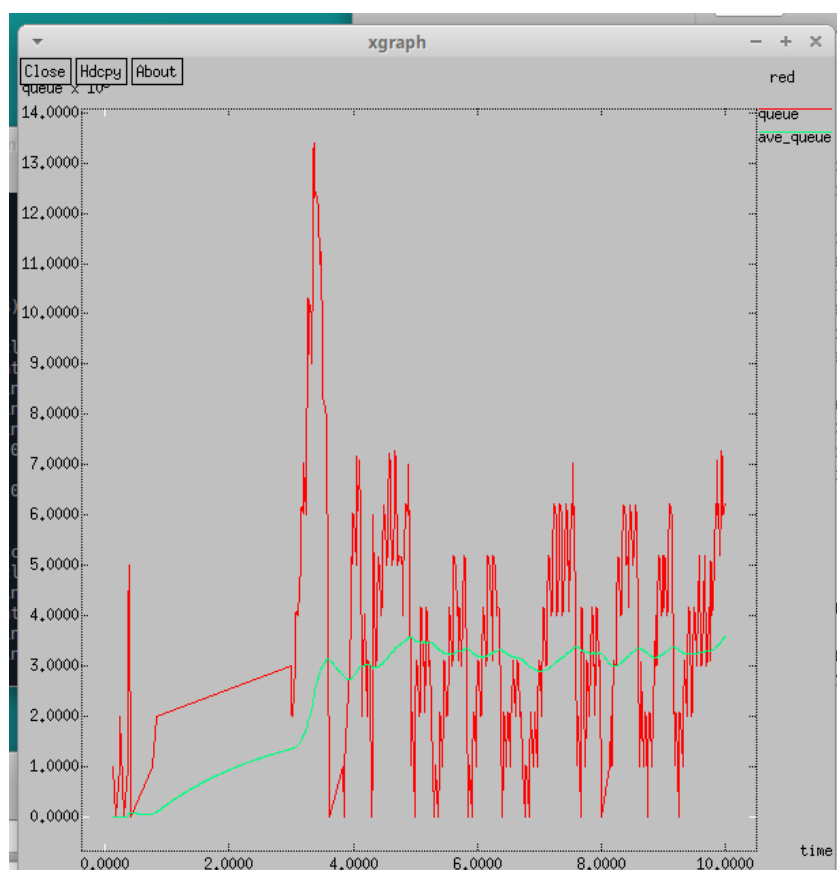


**Рис. 4.4: Изменение TCP-окна.**

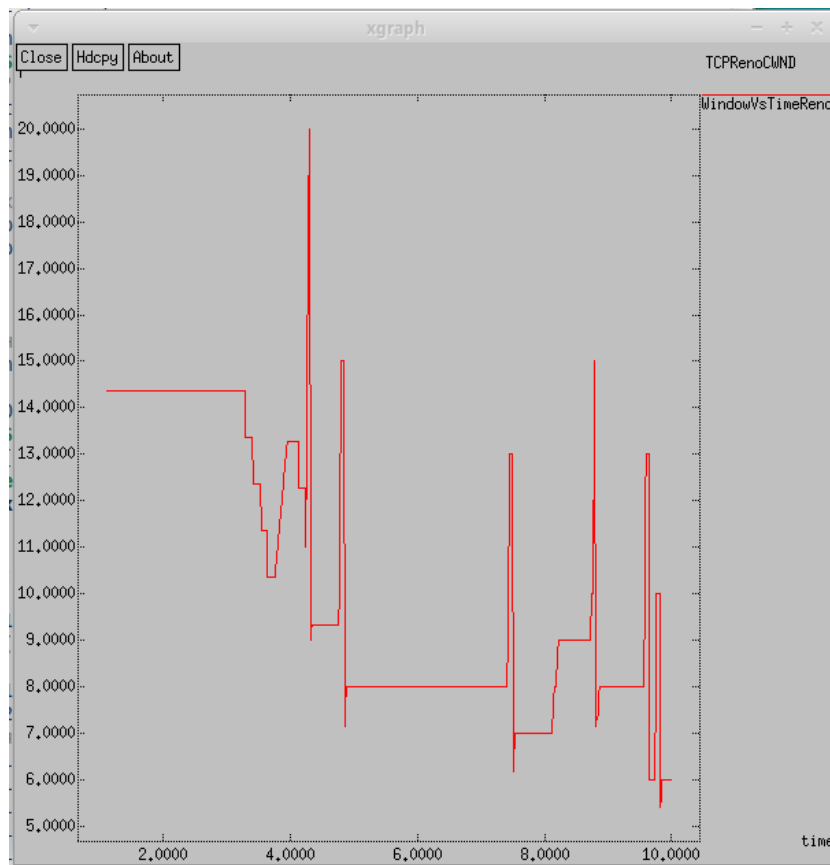
Далее, следуя заданию, изменим Newreno для s1 на Vegas. Для этого изменила следующую часть кода:

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

Получились следующие графики (рис. [4.5] и [4.6]):



**Рис. 4.5: Изменение длины очереди.**



**Рис. 4.6: 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 \"TCPReoCWND\" WindowVsTimeReno &
exec xgraph -fg white -bg pink -bb -tk -x time -y ochered temp.queue &
exit 0

```

**Получились следующие графики (рис. [4.7] и [4.8]):**



**Рис. 4.7: Изменение длины очереди.**



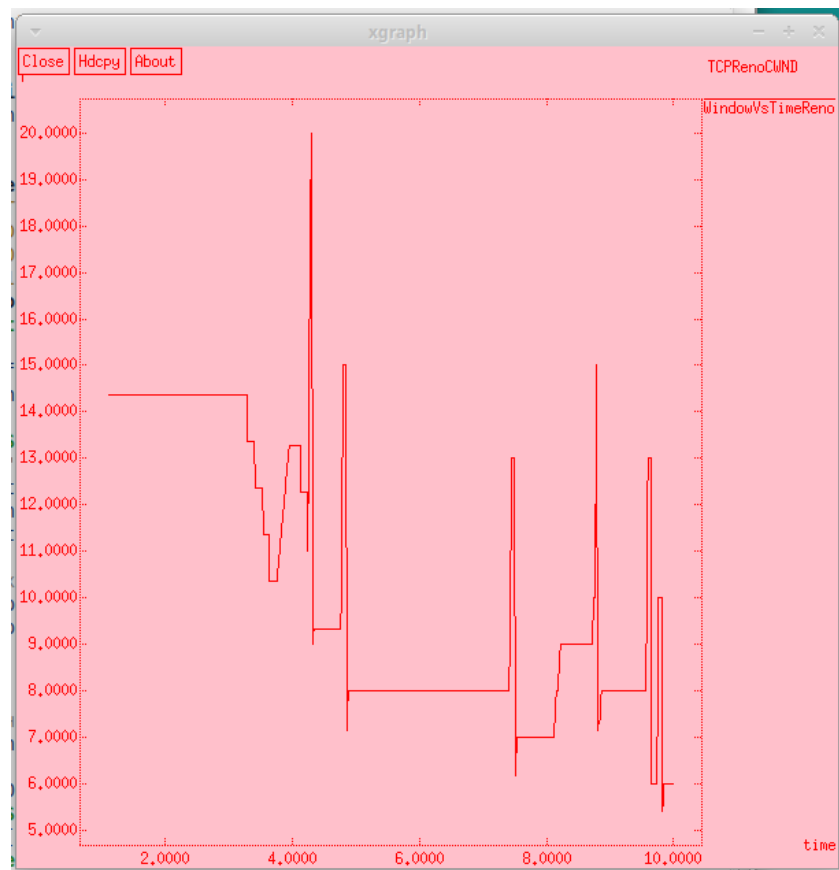


Рис. 4.8: Изменение TCP-окна.

## **5 Выводы**

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