

Лабораторная работа 2

Тагиев Байрам Алтай оглы

Содержание

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

Список иллюстраций

3.1	Reno - Размер окна TCP	8
3.2	Reno - Длина очереди и средняя длина очереди	9
3.3	NewReno - Размер окна TCP	9
3.4	NewReno - Длина очереди и средняя длина очереди	9
3.5	Vegas - Размер окна TCP	10
3.6	Vegas - Длина очереди и средняя длина очереди	10

Список таблиц

1 Цель работы

Получение навыков по работе с NS2, моделирование сети с применением RED.

2 Задание

Постановка задачи 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;
- ТСР-источники на узлах s1 и s2 подключаются к ТСР-приёмнику на узле s3;
- генераторы трафика FTP прикреплены к ТСР-агентам.

3 Выполнение лабораторной работы

1. Создадим наши 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]
```

2. Зададим соединения между узлами.

```
$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
```

3. Добавим агентов 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]

```

4. Добавим мониторинг размера окна.

```

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;

```

5. Также добавим 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_

```

6. Добавим дополнительную процедуру для формирования файла с данными о размере окна TCP. Не самое

7. И получим следующие графики.

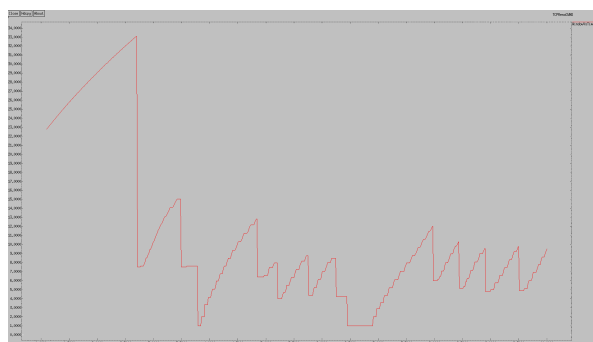


Рис. 3.1: Reno - Размер окна TCP

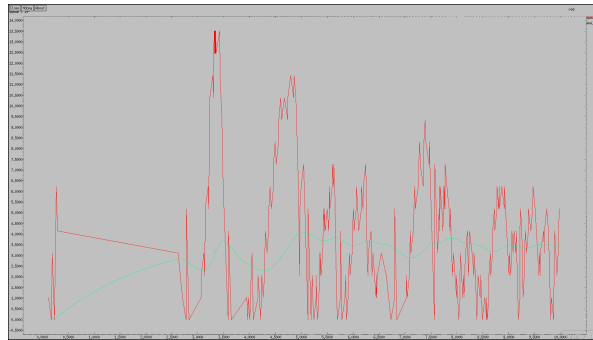


Рис. 3.2: Reno - Длина очереди и средняя длина очереди

8. Давайте проверим другие типы протоколов. Начнем с NewReno. Работает немного эффективнее, но схожа с Reno.

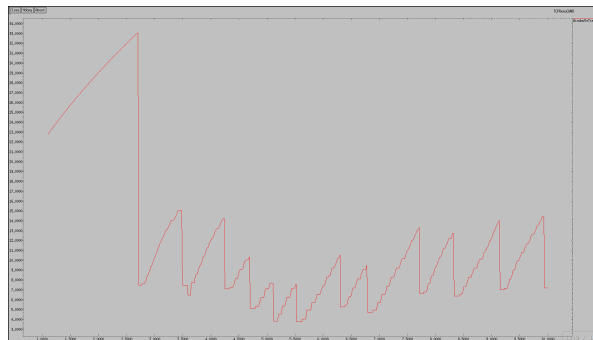


Рис. 3.3: NewReno - Размер окна TCP

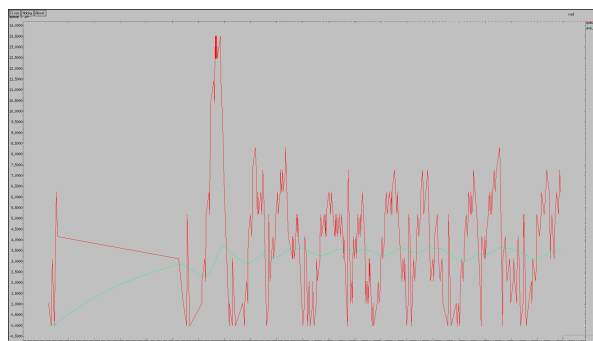


Рис. 3.4: NewReno - Длина очереди и средняя длина очереди

9. Но мы можем еще улучшить наши результаты, применив другого тип протокола TCP - Vegas.

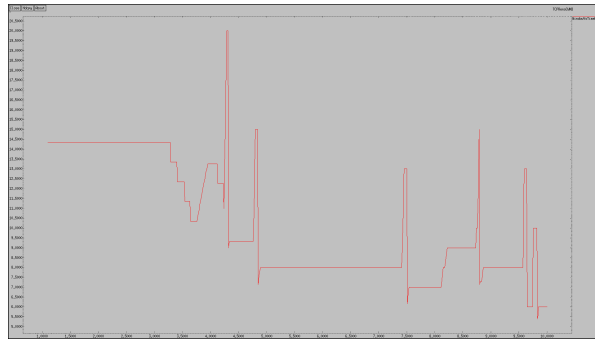


Рис. 3.5: Vegas - Размер окна TCP

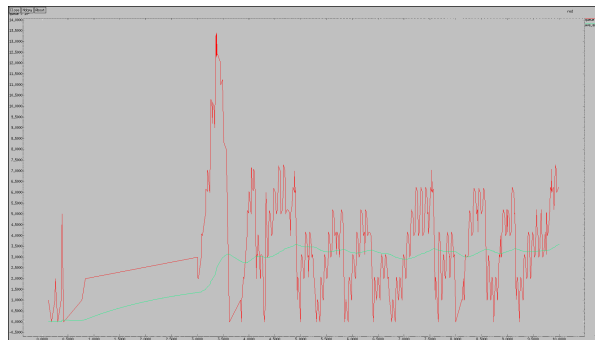


Рис. 3.6: Vegas - Длина очереди и средняя длина очереди

10. Если нам понадобится поменять как-то кривую, то можно изменять параметры `xgraph`, который используется для отрисовки графиков.

<https://manpages.ubuntu.com/manpages/xenial/man1/xgraph.1.html>

4 Выводы

По мере выполнения работы, я приобрел практические навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также смоделировал сеть с RED.