Лабораторная работа № 2

Имитационное моделирование

Королёв Иван Андреевич

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Более подробно познакомится с протоколом TCP и мониторингом очередей.

# 2 Задание

# 3 Пример с дисциплиной RED

* 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.

# 4 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).

# 5 Теоретическое введение

# 6 Протокол TCP

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

В ns-2 поддерживает следующие TCP-агенты односторонней передачи: \* Agent/TCP \* Agent/TCP/Reno \* Agent/TCP/Newreno \* Agent/TCP/Sack1 — TCP с выборочным повтором (RFC2018) \* Agent/TCP/Vegas \* Agent/TCP/Fack — Reno TCP с «последующим подтверждением» \* Agent/TCP/Linux — TCP-передатчик с поддержкой SACK, который использует TCP с перезагрузкой контрольных модулей из ядра Linux Односторонние агенты приёма: \* Agent/TCPSink \* Agent/TCPSink/DelAck \* Agent/TCPSink/Sack1 \* Agent/TCPSink/Sack1/DelAck Двунаправленный агент: \* Agent/TCP/FullTcp

# 7 Мониторинг очередей

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

Объекты очереди: \* qlim\_ — максимально разрешённое число пакетов в очереди; \* limit\_ — размер очереди в пакетах; \* blocked\_ — принимает значение true, если очередь заблокирована; \* unblock\_on\_resume\_ — принимает значение true, указывая, что очередь должна быть разблокирована после отправки последнего пакета; \* bytes\_ — принимает значение true, если используется режим передачи в байтах, а не в пакетах; \* queue-in-bytes\_ — принимает значение true, если используется режим измерения среднего размера очереди в байтах, а не пакетах; \* thresh\_ — минимальный порог среднего размера очереди (в пакетах); \* maxthresh\_ — максимальный порог среднего размера очереди (в пакетах); \* mean\_pktsize\_ — грубая оценка среднего размера пакета (в байтах); \* q\_weight\_ — вес очереди (используется при расчёте экспоненциальновзвешенного скользящего среднего размера очереди; \* wait\_ — интервал времени между сброшенными пакетами. Объекты мониторинга очереди: \* size\_ — размер мгновенной длины очереди (в байтах); \* pkts\_ — размер мгновенной длины очереди (в пакетах); \* parrivals\_ — промежуточная сумма поступивших пакетов; \* barrivals\_ — промежуточная сумма байт в поступивших пакетах \* pdepartures\_ — промежуточная сумма обслуженных пакетов (не отброшенных); \* bdepartures\_ — промежуточная сумма байт обслуженных пакетов (не отброшенных); \* pdrops\_ — общая сумма отброшенных пакетов; \* bdrops\_ — общая сумма байт отброшенных пакетов; \* bytesInt\_ — заполненность очереди в байтах; \* pktsInt\_ — заполненность очереди в пакетах; \* epdrops\_ — число сброшенных по алгоритму RED пакетов; \* ebdrops\_ — число байт в сброшенных по алгоритму RED пакетах; \* enable\_in\_ — устанавливается значение true, если требуется мониторинг потока на входе; \* enable\_out\_ — устанавливается значение true, если требуется мониторинг потока на выходе; \* enable\_drop\_ — устанавливается значение true, если требуется мониторинг сброшенных из потока пакетов; \* enable\_edrop\_ — устанавливается значение true, если требуется мониторинг сброшенных из потока пакетов по алгоритму RED; \* src\_ — адрес источника пакетов, принадлежащих потоку; \* dst\_ — адрес получателя пакетов, принадлежащих потоку; \* flowid\_ — идентификатор потока.

# 8 Выполнение лабораторной работы

## 8.1 Пример с дисциплиной RED

### 8.1.1 Реализация модели

Реализация модели. Описываются узлы сети, соединения, агенты и приложения, мониторинг размера окна, мониторинг очереди, добавление at-событий и формирование файла с данными о размере окна TCP. (рис. 1).

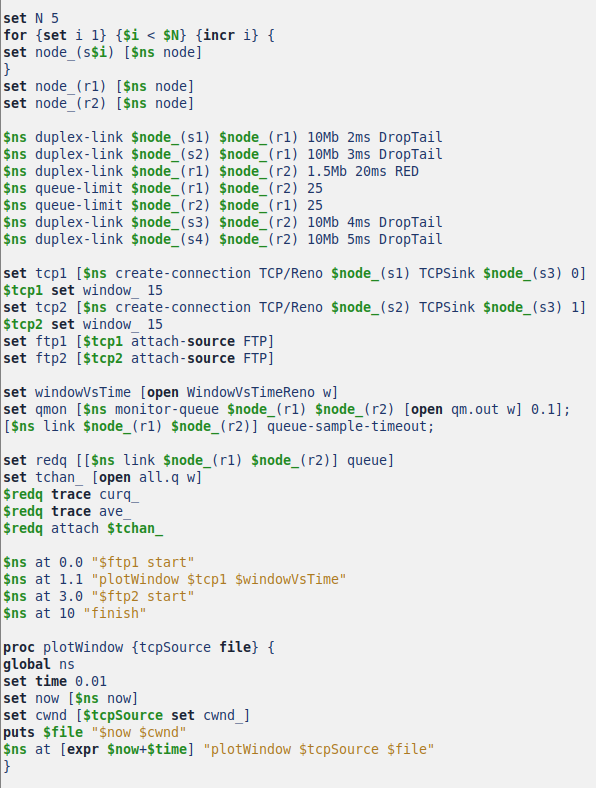


Рис. 1: Реализация модели

Реализация модели. Добавление процедуры finish. Подключение кода AWK. Открытие файла f на запись. Выполнение кода AWK, подпись траекторий в легенде. Запуск xgraph с графиками окна TCP и очереди. (рис. 2).



Рис. 2: Реализация модели

### 8.1.2 График изменения TCP-окна. Тип TCP-агента: Reno

В начале соединения окно перегрузки быстро растет, что характерно для фазы медленного старта. Примерно на 2 секундах происходит резкое уменьшение окна, что свидетельствует о потери пакетов. После первого спада окно перегрузки начинает увеличиваться, но с периодическими падениями, что указывает на Reno. Максимальное значение окна примерно 33, минимальное значение окна около 1. (рис. 3).

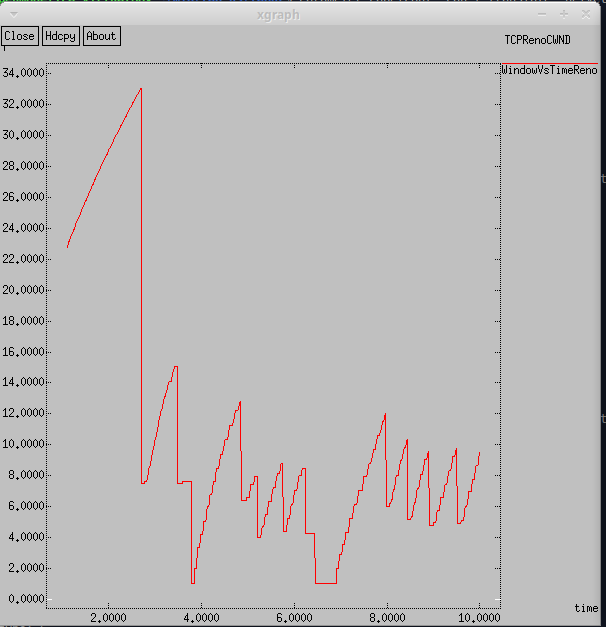


Рис. 3: График изменения TCP-окна. Тип TCP-агента: Reno

### 8.1.3 График изменения длины очереди и средней длины очереди. Тип TCP-агента: Reno

Текущий размер очереди показывает высокие колебания. Средняя очередь постепенно растет и остается на стабильном уровне, что говорит о настройке RED. Максимальное значение примерно 13.5, минимальное значение окна около 0. (рис. 4).

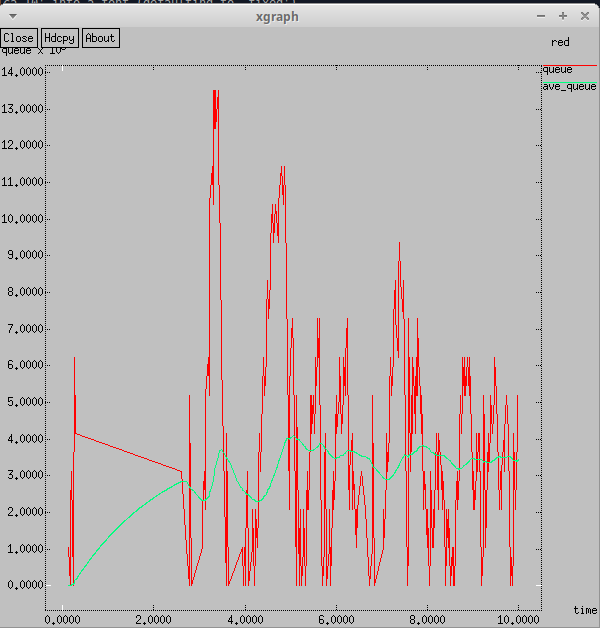


Рис. 4: График изменения длины очереди и средней длины очереди. Тип TCP-агента: Reno

## 8.2 Упражнение.

### 8.2.1 Изменить в модели на узле s1 тип протокола TCP с Reno на NewReno.

Изменяем тип протокола на NewReno.(рис. 5).

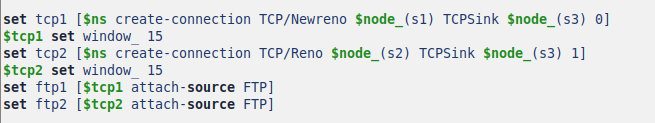


Рис. 5: Модель. Изменили на другой тип протокола.

### 8.2.2 График изменения TCP-окна. Тип TCP-агента: NewReno

В начале соединения окно перегрузки быстро растет, что характерно для фазы медленного старта. Примерно на 3 секундах происходит резкое уменьшение окна, что свидетельствует о потери пакетов. После первого спада окно перегрузки начинает увеличиваться, но с периодическими падениями, что указывает на Reno. Максимальное значение окна примерно 33, минимальное значение окна около 4. (рис. 6).

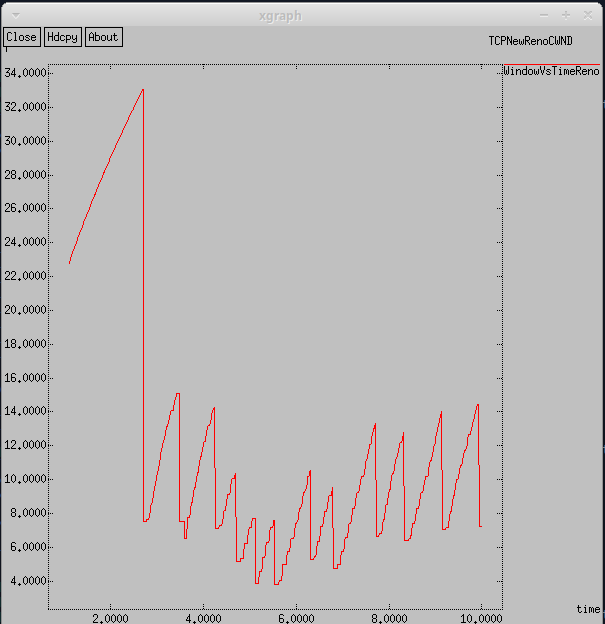


Рис. 6: График изменения TCP-окна. Тип TCP-агента: NewReno

### 8.2.3 График изменения длины очереди и средней длины очереди. Тип TCP-агента: NewReno

Текущий размер очереди показывает довольно высокие колебания, но более маленькие, чем у Reno. Средняя очередь постепенно растет и остается на стабильном уровне, что говорит о настройке RED. Максимальное значение примерно 13.5, минимальное значение окна около 0. (рис. 7).

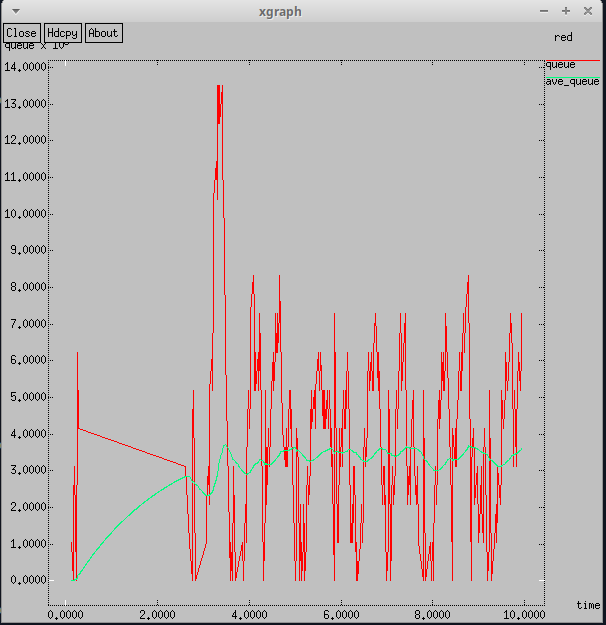


Рис. 7: График изменения длины очереди и средней длины очереди. Тип TCP-агента: NewReno

### 8.2.4 Изменить в модели на узле s1 тип протокола TCP с Reno на Vegas

Изменяем тип протокола на Vegas.(рис. 8).

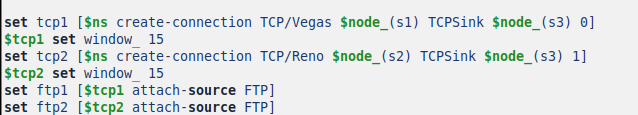


Рис. 8: Модель. Изменили на другой тип протокола.

### 8.2.5 График изменения TCP-окна. Тип TCP-агента: Vegas

В начале соединения окно перегрузки сохраняет значение.При Vegas максимальный размер окна составляет 20, а не 34, как в NewReno. Vegas обнаруживает перегрузки сети до того, как произойдет потеря, поэтому можно сказать он не теряет пакеты. (рис. 9).

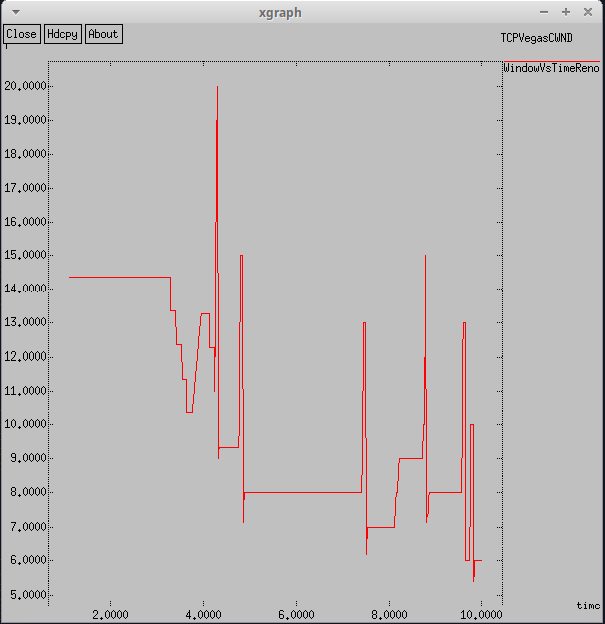


Рис. 9: График изменения TCP-окна. Тип TCP-агента: Vegas

### 8.2.6 График изменения длины очереди и средней длины очереди. Тип TCP-агента: Vegas

С нуля до примерно двух секунд, происходит скачок и плавное изменение значения в текущем размере очереди. Текущий размер очереди показывает довольно высокие колебания. Средняя очередь постепенно растет и остается на стабильном уровне, что говорит о настройке RED. Максимальное значение примерно 13.5, минимальное значение окна около 5. (рис. 10).

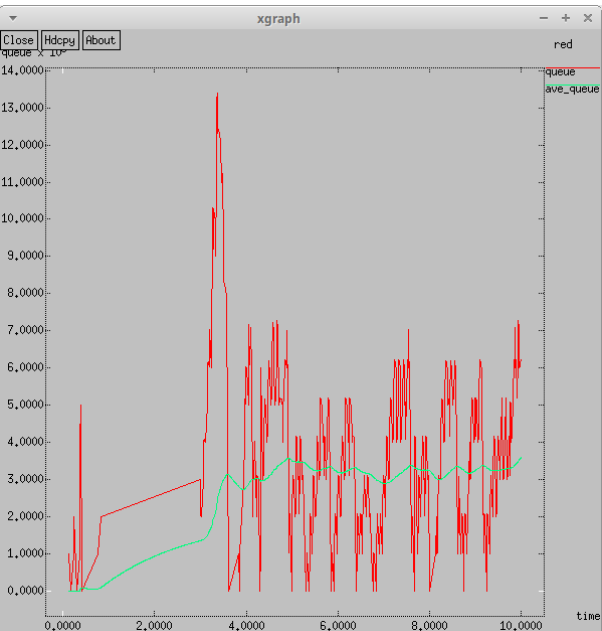


Рис. 10: График изменения длины очереди и средней длины очереди. Тип TCP-агента: Vegas

### 8.2.7 Сравнение Newreno и Vegas

* Newreno лучше для сетей с высокой потерей пакетов.
* Vegas эффективнее в условиях высокой задержки.

Если сеть имеет переменную задержку и много конкурирующих соединений, то лучше выбрать Newreno. Если же сеть стабильна и перегруженность предсказуема, то Vegas.

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

Сначала изменяю цвет фона и подписи к осям. После, добавляю новые подписи траекторий в легенде и цвет траекторий. (рис. 11), (рис. 12)

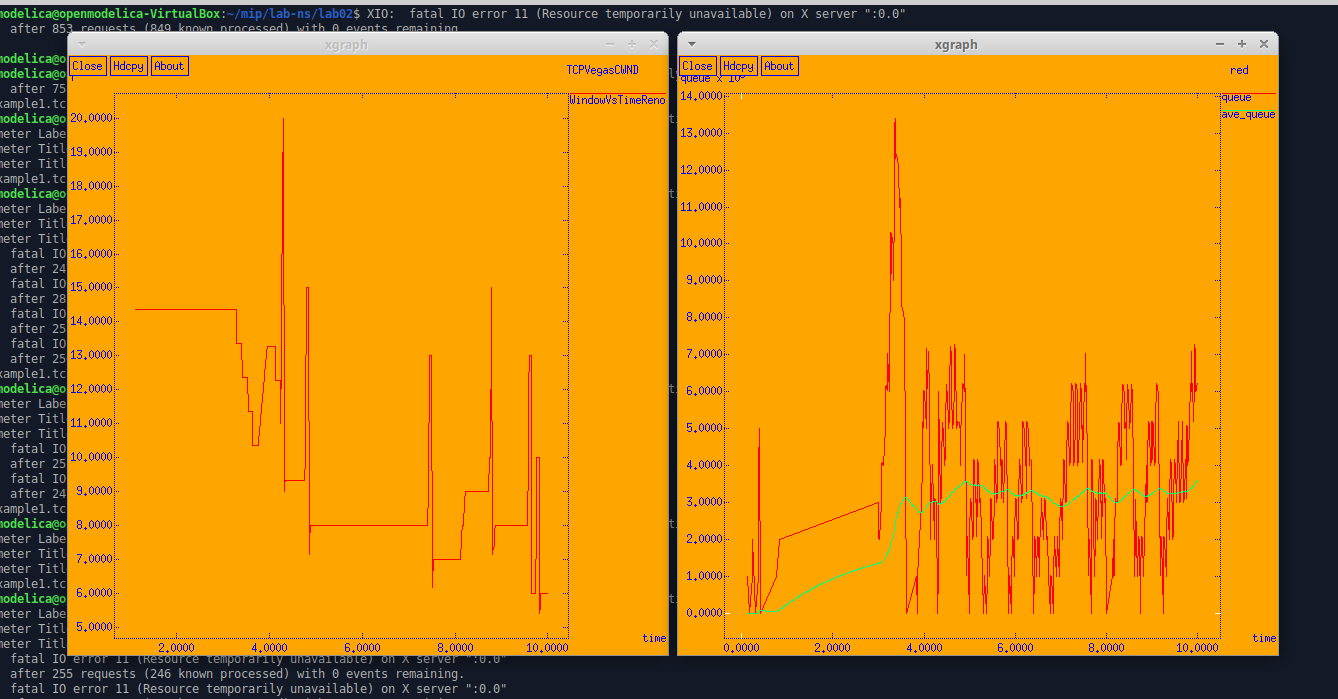


Рис. 11: Цвет фона и подписи к осям



Рис. 12: Цвет траектории и подписи траектории в легенде

Измененный код реализации модели. (рис. 13), (рис. 14)

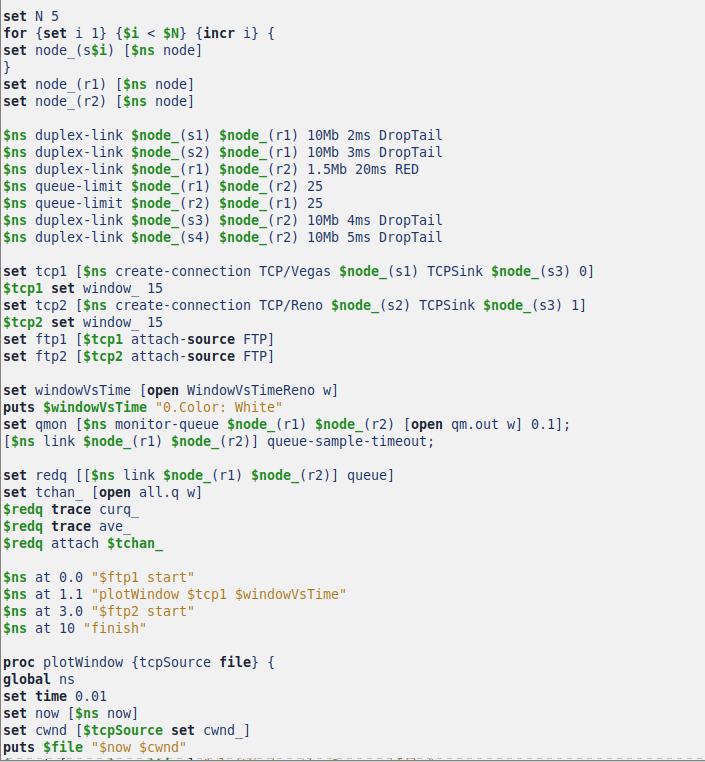


Рис. 13: Реализация

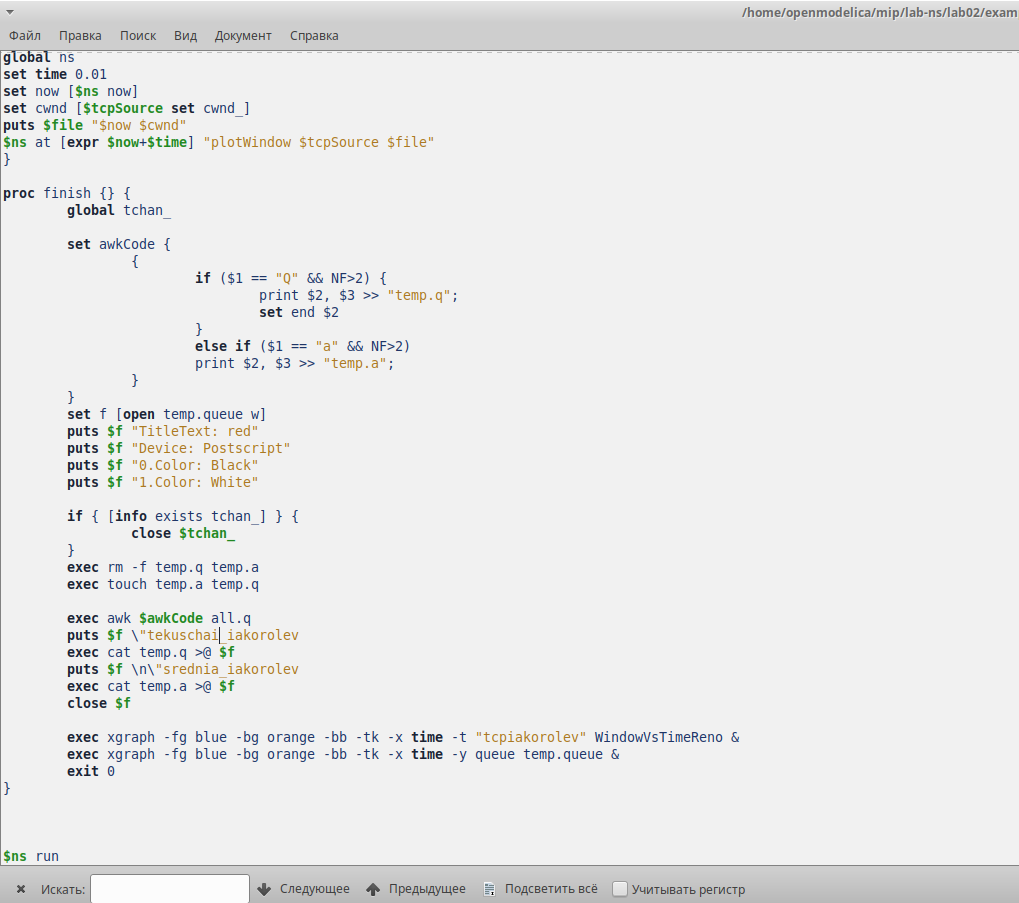


Рис. 14: Реализация

# 9 Выводы

Более подробно познакомился с протоколом TCP и мониторингом очередей.

# Список литературы