Лабораторная работа 2

Исследование протокола TCP и алгоритма управления очередью RED

Извекова Мария Петровна

Содержание

Список иллюстраций

Список таблиц

# Цель работы

Ознакомиться с протоколом TCP и очередью RED, построить сценарий на симмуляторе и изобразить результате в Xgraph

# Задание

Требуется разработать сценарий, реализующий модель из 6 узлов с дуплексным соединением и очередью c дисциплиной RED, построить в Xgraph график изменения TCP-окна, график изменения длины очереди и средней длины очереди

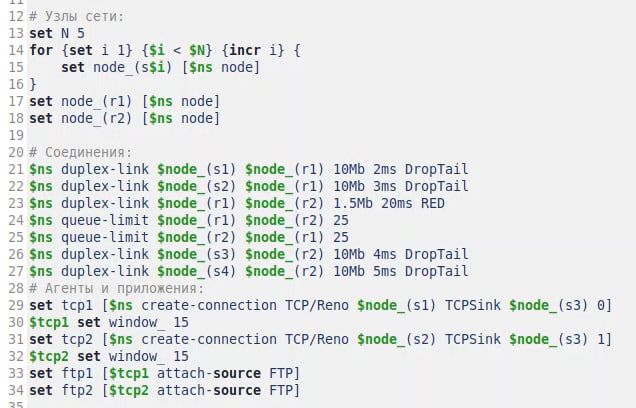
# Теоретическое введение

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

Флаг Указатель срочности (Urgent Pointer, URG) устанавливается в 1 в случае использования поля Указатель на срочные данные. Флаг Подтверждение (Acknowledgment, ACK) устанавливается в 1 в случае, если поле Номер подтверждения (Acknowledgement Number) содержит данные. В противном случае это поле игнорируется. Флаг Выталкивание (Push, PSH) означает, что принимающий стек TCP должен немедленно информировать приложение о поступивших данных, а не ждать, пока буфер заполниться. Флаг Сброс (Reset, RST) используется для отмены соединения из-за ошибки приложения, отказа от неверного сегмента, попытки создать соединение при отсутствии затребованного сервиса. Флаг Синхронизация (Synchronize, SYN) устанавливается при инициировании соединения и синхронизации порядкового номера. Флаг Завершение (Finished, FIN) используется для разрыва соединения. Он указывает, что отправитель закончил передачу данных. Управление потоком в протоколе TCP осуществляется при помощи скользящего окна переменного размера: – поле Размер окна (Window) (длина 16 бит) содержит количество байт, которое может быть послано после байта, получение которого уже подтверждено; – если значение этого поля равно нулю, это означает, что все байты, вплоть до байта с номером Номер подтверждения - 1, получены, но получатель отказывается принимать дальнейшие данные; – разрешение на дальнейшую передачу может быть выдано отправкой сегмента с таким же значением поля Номер подтверждения и ненулевым значением поля Размер окна. Регулирование трафика в TCP: – контроль доставки — отслеживает заполнение входного буфера получателя с помощью параметра Размер окна (Window); – контроль перегрузки — регистрирует перегрузку канала и связанные с этим потери, а также понижает интенсивность трафика с помощью Окна перегрузки (Congestion Window, CWnd) и Порога медленного старта (Slow Start Threshold, SSThreth)

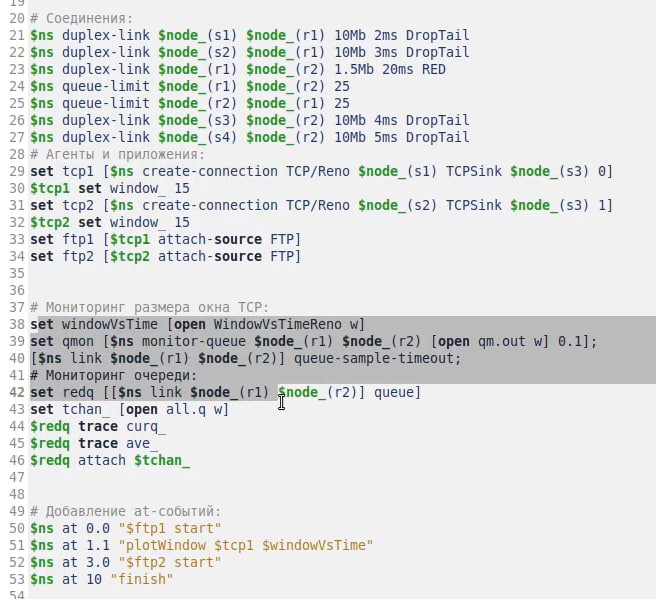
# Выполнение лабораторной работы

1. Создаем скрипт где прописываем все соединения 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.:



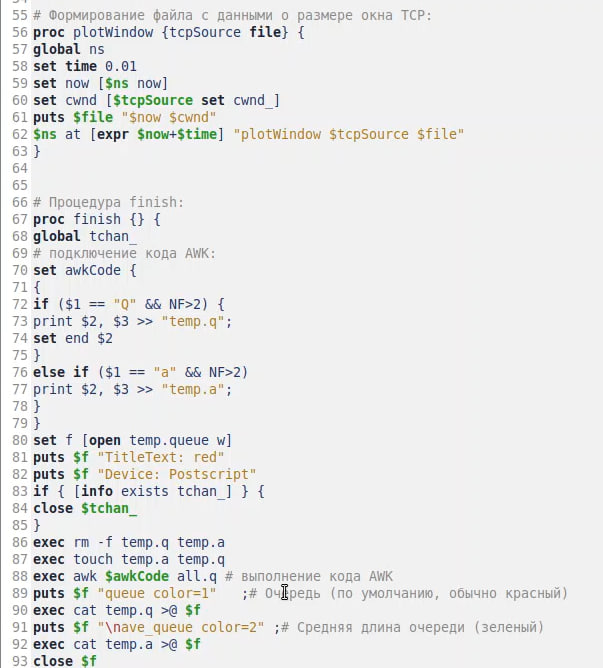
Создание сети

1. В скрипте прописываем мониторинг размера окна и очереди – то, что исследуем



Мониторинг окна

1. Прописываем часть, отвечающая за вывод результата в Xgraph. результаты их временных таблиц переносим в temp.queue



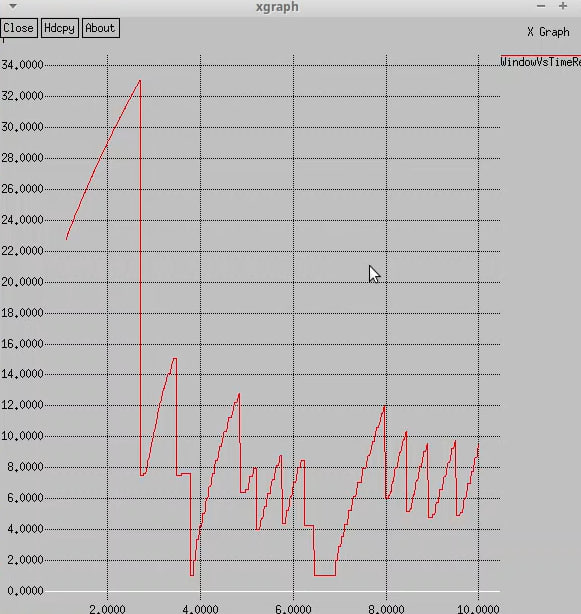
Скрипт Xgraph

1. Сначала рассматриваем протокол TCP Reno

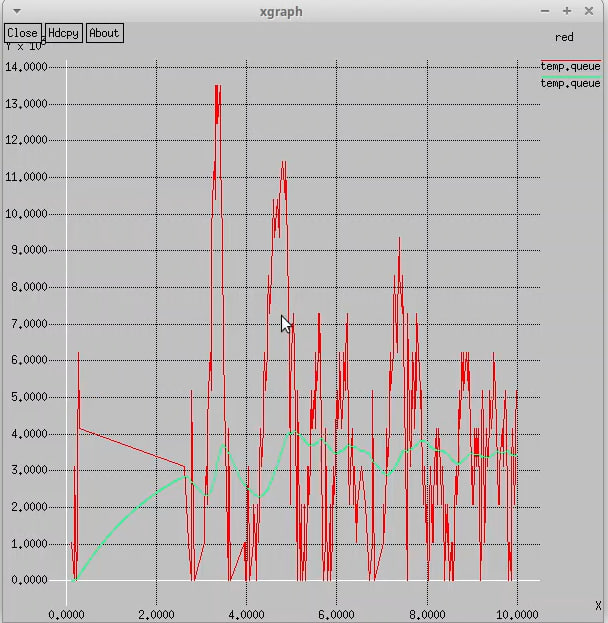
Протоколы

Протоколы

1. Результаты размеров окна и очереди



Динамика размеры окна



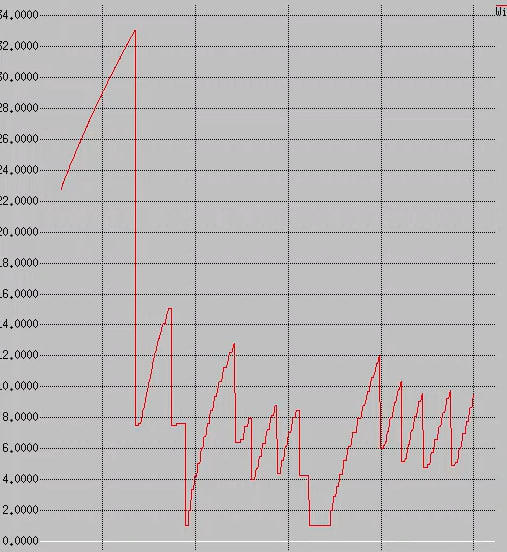
Мониторинг очереди

1. Рассматриваем протокол TCP NewReno

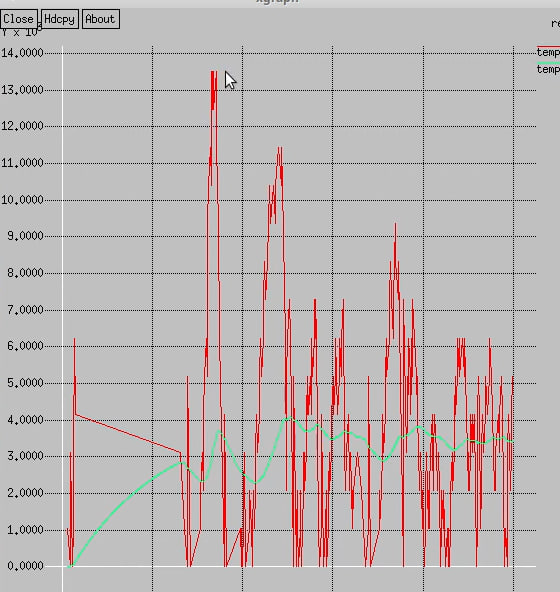
Протоколы NewReno

Протоколы NewReno

1. Результаты размеров окна и очереди



Динамика размеры окна



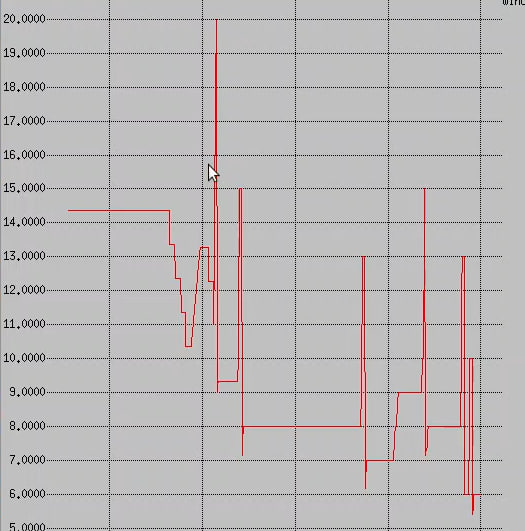
Мониторинг очереди

1. Рассматриваем протокол TCP Vegas

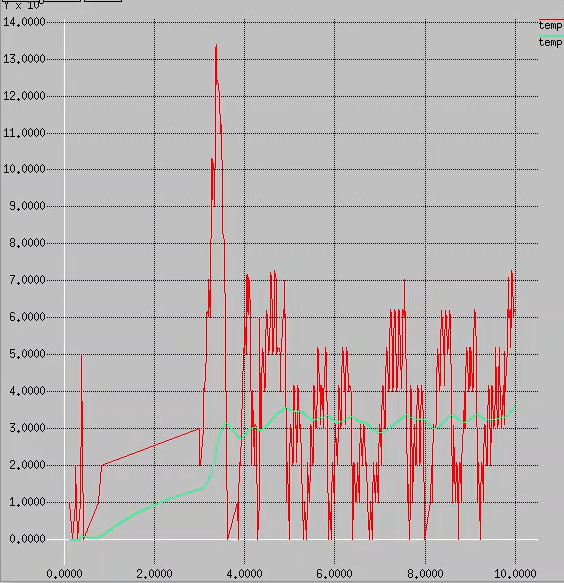
Протоколы Vegas

Протоколы Vegas

1. Результаты размеров окна и очереди



Динамика размеры окна



Мониторинг очереди

# Выводы

Ознакомиться с протоколом TCP и очередью RED, построить сценарий на симмуляторе и изобразить результате в Xgraph. Так же мы рассмотрели, что протоколы TCP Reno и NewReno реагируют на изменения, только после потерей пакетов, как протокол Vegas замечает это до, из-за чего его линии моментами не меняют направление.

# Список литературы