Лабораторная работа №1

**Простые модели компьютерной сети**

Игнатенкова Варвара Николаевна

# 1 Цель работы

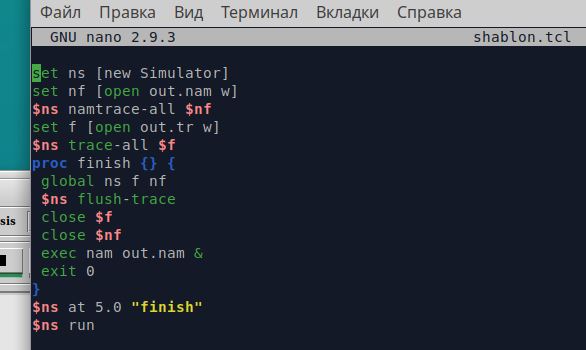
Приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

# 2 Теоретическое введение

Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года; координатор — Дзюн Хамано [1].

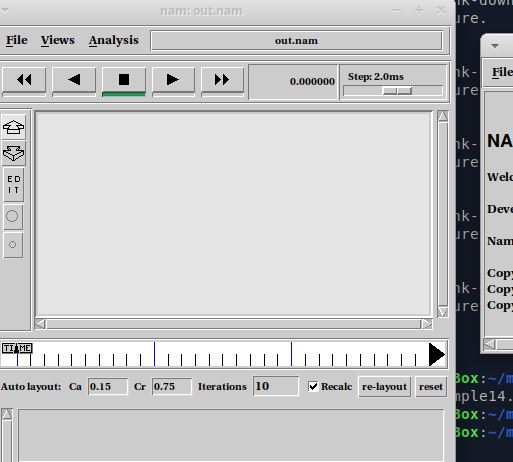
# 3 Выполнение лабораторной работы

## 3.1 Шаблон сценария для NS-2

В своём рабочем каталоге создадим директорию mip, к которой будут выполняться лабораторные работы. Внутри mip создадим директорию lab-ns, а в ней файл shablon.tcl.

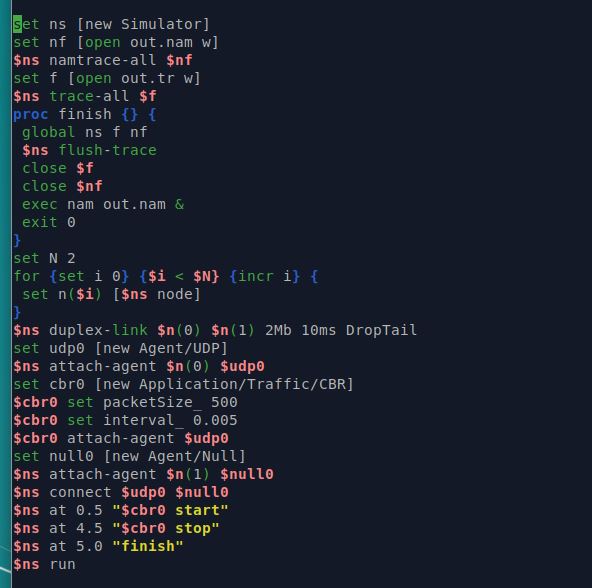
shablon.tcl

Сохранив изменения в отредактированном файле shablon.tcl и закрыв его, можно запустить симулятор командой: ns shablon.tcl

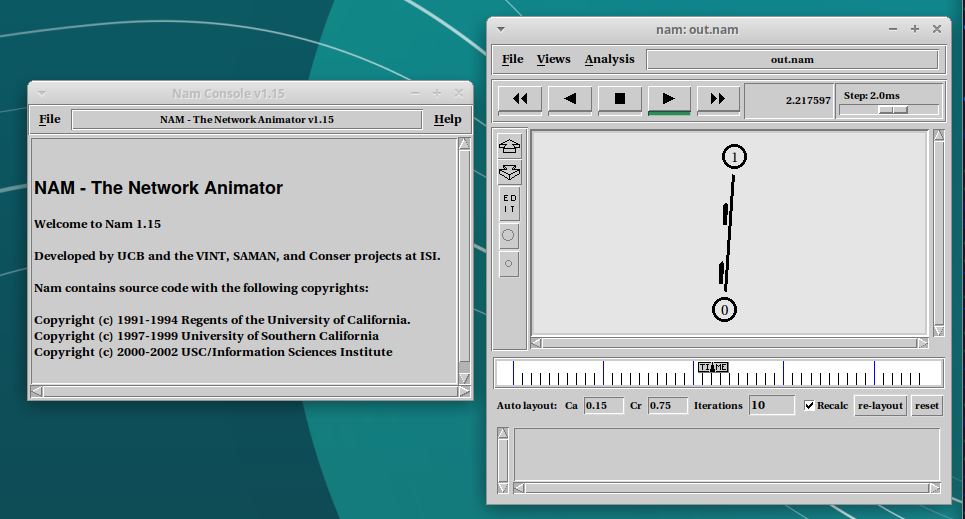


При этом на экране появится сообщение типа nam: empty trace file out.nam поскольку ещё не определены никакие объекты и действия.

## 3.2 Простой пример описания топологии сети, состоящей из двух узлов и одного соединения

**Постановка задачи.** Требуется смоделировать сеть передачи данных, состоящую из двух узлов, соединённых дуплексной линией связи с полосой пропускания 2 Мб/с и задержкой 10 мс, очередью с обслуживанием типа DropTail. От одного узла к другому по протоколу UDP осуществляется передача пакетов, размером 500 байт, с постоянной скоростью 200 пакетов в секунду.

Реализация модели

При нажатии на кнопку play в окне nam через 0.5 секунды из узла 0 данные начнут поступать к узлу 1. Это процесс можно замедлить, выбирая шаг отображения в nam. Можно осуществлять наблюдение за отдельным пакетом, щёлкнув по нему в окне nam, а щёлкнув по соединению, можно получить о нем некоторую информацию.

## 3.3 Пример с усложнённой топологией сети

**Постановка задачи.** Описание моделируемой сети (рис. 2.4):– сеть состоит из 4 узлов (n0, n1, n2, n3);

– между узлами n0 и n2, n1 и n2 установлено дуплексное соединение с пропускной способностью 2 Мбит/с и задержкой 10 мс;

– между узлами n2 и n3 установлено дуплексное соединение с пропускной способностью 1,7 Мбит/с и задержкой 20 мс;

– каждый узел использует очередь с дисциплиной Drop Tail для накопления пакетов, максимальный размер которой составляет 10;

– TCP-источник на узле n0 подключается к TCP-приёмнику на узле n3

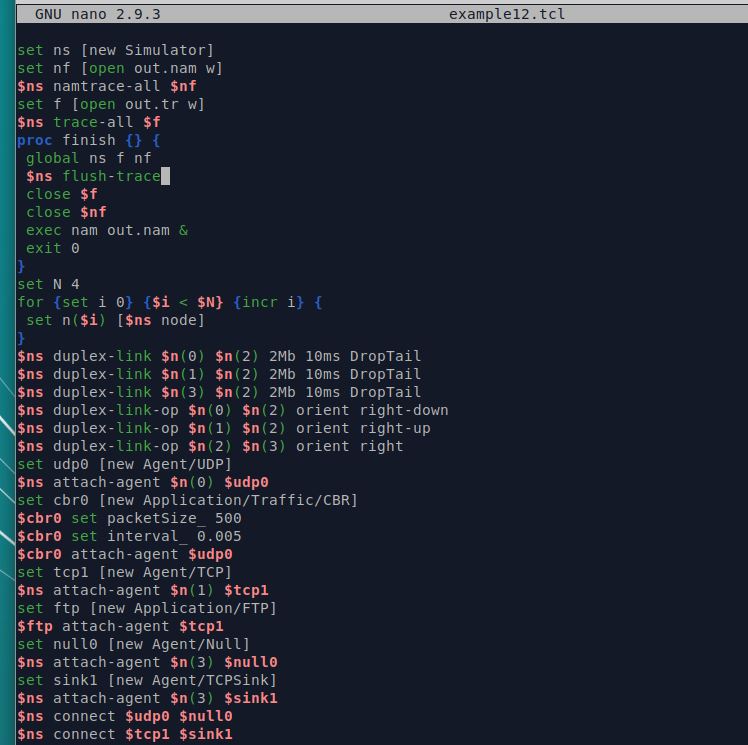
(по-умолчанию, максимальный размер пакета, который TCP-агент может генерировать, равняется 1KByte)

– TCP-приёмник генерирует и отправляет ACK пакеты отправителю и откидывает полученные пакеты;

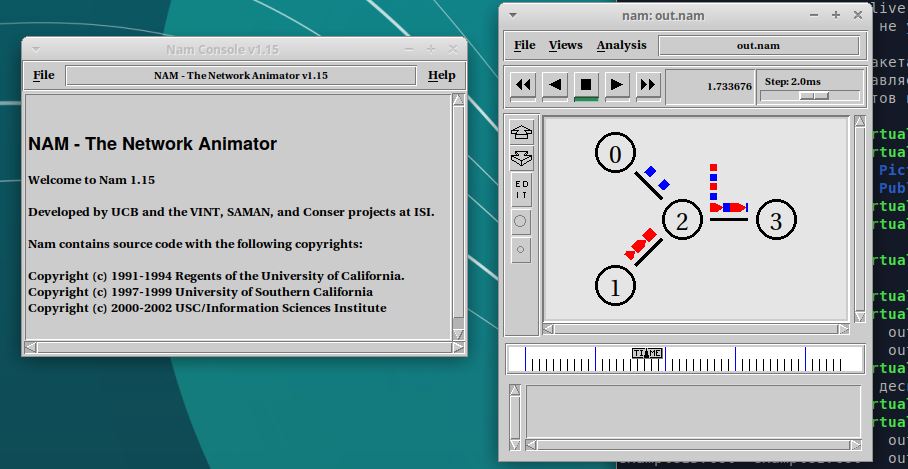
– UDP-агент, который подсоединён к узлу n1, подключён к null-агенту на узле n3 (null-агент просто откидывает пакеты);

– генераторы трафика ftp и cbr прикреплены к TCP и UDP агентам соответственно;

– генератор cbr генерирует пакеты размером 1 Кбайт со скоростью 1 Мбит/с;

– работа cbr начинается в 0,1 секунду и прекращается в 4,5 секунды, а ftp начинает работать в 1,0 секунду и прекращает в 4,0 секунды.

Реализация модели

При запуске скрипта можно заметить, что по соединениям между узлами n(0)–n(2) и n(1)–n(2) к узлу n(2) передаётся данных больше, чем способно передаваться по соединению от узла n(2) к узлу n(3). Действительно, мы передаём 200 пакетов в секунду от каждого источника данных в узлах n(0) и n(1), а каждый пакет имеет размер 500 байт. Таким образом, полоса каждого соединения 08 Mb, а суммарная —16Mb.Носоединение n(2)–n(3) имеет полосу лишь 1 Mb. Следовательно, часть пакетов должна теряться. В окне аниматора можно видеть пакеты в очереди, а также те пакеты, которые отбрасываются при переполнении.

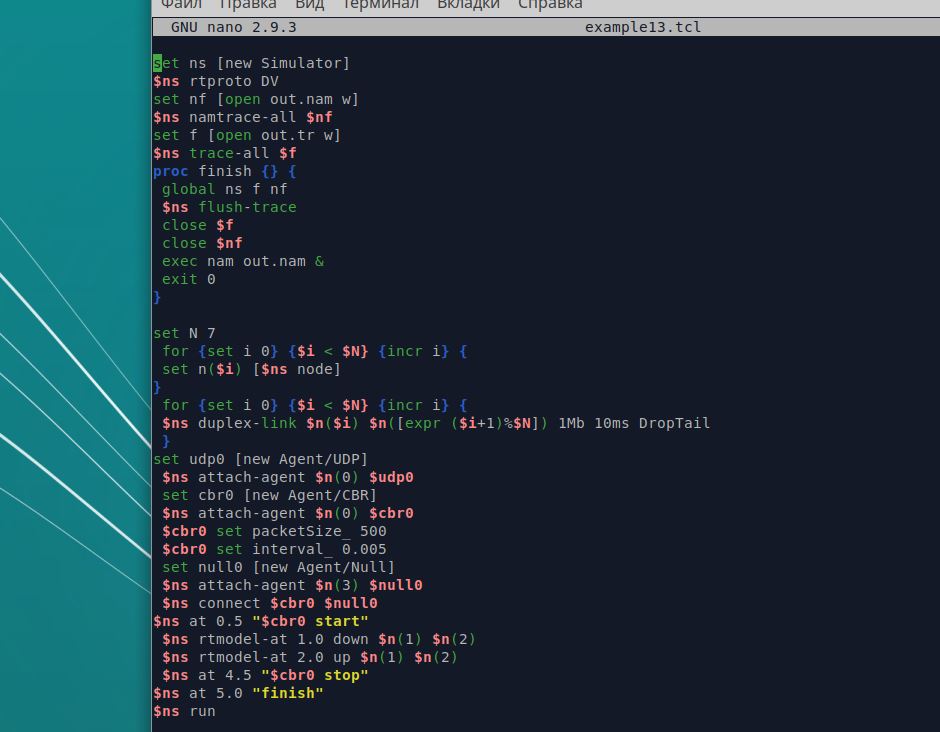
# 3.4 Пример с кольцевой топологией сети

**Постановка задачи.** Требуется построить модель передачи данных по сети с кольцевой топологией и динамической маршрутизацией пакетов:

– сеть состоит из 7 узлов, соединённых в кольцо;

– данные передаются от узла n(0) к узлу n(3) по кратчайшему пути;

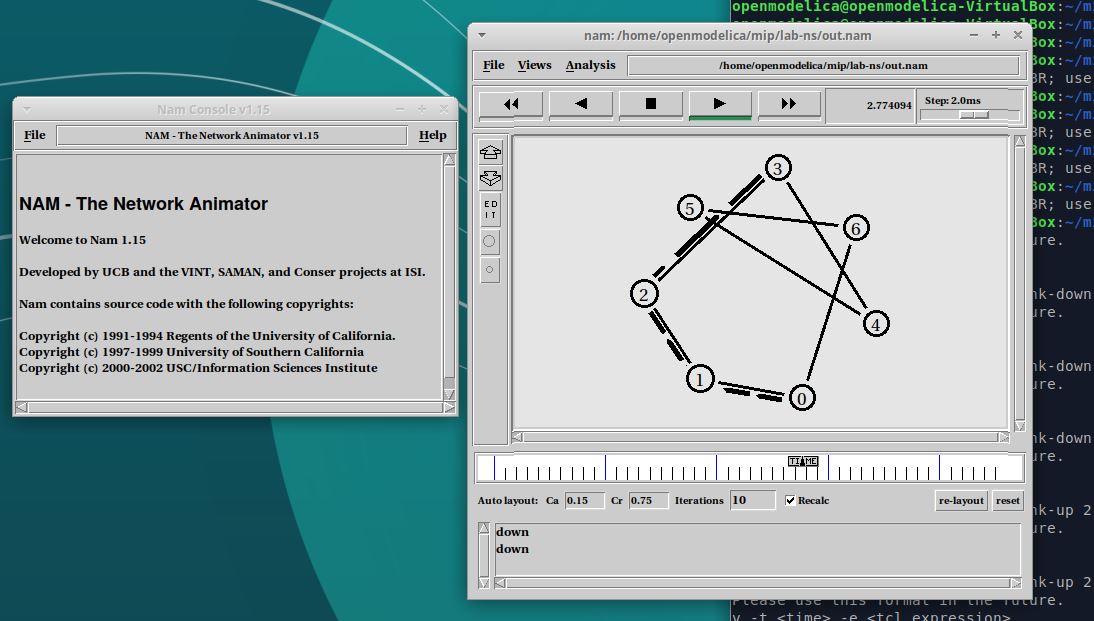
– с 1 по 2 секунду модельного времени происходит разрыв соединения между узлами n(1) и n(2);

– при разрыве соединения маршрут передачи данных должен измениться на резервный.

Реализация модели

Сразу после запуска в сети отправляется небольшое количество маленьких пакетов, используемых для обмена информацией, необходимой для маршрутизации между узлами. Когда соединение будет разорвано, информация о топологии будет обновлена, и пакеты будут отсылаться по новому маршруту через узлы n(6), n(5) и n(4).

# 3.5 Упражнение

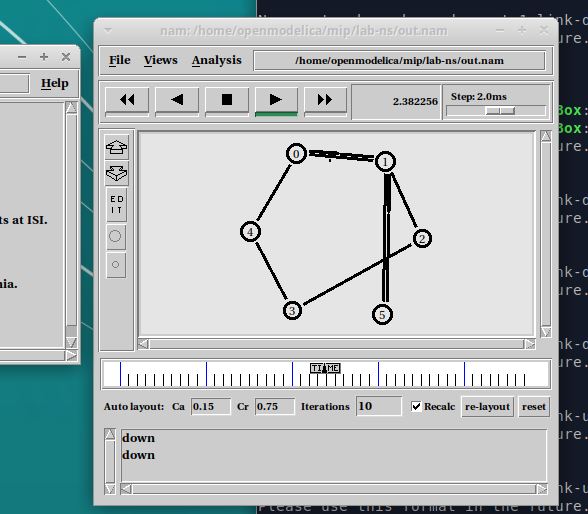
**Постановка задачи.** Внесите следующие изменения в реализацию примера с кольцевой топологией сети:

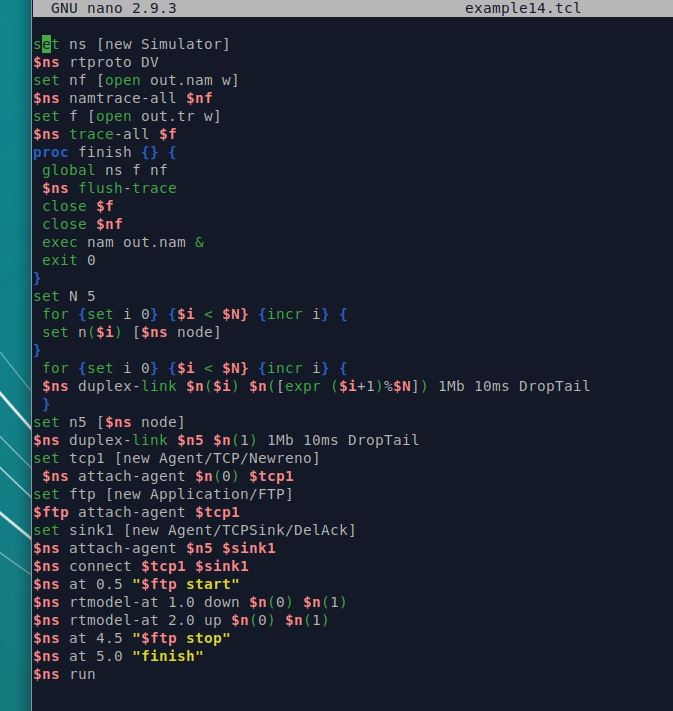
– топология сети должна соответствовать представленной на рис.

– передача данных должна осуществляться от узла n(0) до узла n(5) по кратчайшему пути в течение 5 секунд модельного времени;

– передача данных должна идти по протоколу TCP (тип Newreno), на принимаю щей стороне используется TCPSink-объект типа DelAck; поверх TCP работает протокол FTP с 0,5 до 4,5 секунд модельного времени;

– с 1 по 2 секунду модельного времени происходит разрыв соединения между узлами n(0) и n(1);

– при разрыве соединения маршрут передачи данных должен измениться на ре зервный, после восстановления соединения пакеты снова должны пойти по кратчайшему пути.



Реализация модели

Вывод симулятора

# 4 Выводы

Мы приобрели навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2.