Лабораторная работа №1

Простые модели компьютерной сети

Алиева Милена Арифовна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Создать шаблон сценария для NS-2;
2. Выполнить простой пример описания топологии сети, состоящей из двух узлов и одного соединения;
3. Выполнить пример с усложнённой топологией сети;
4. Выполнить пример с кольцевой топологией сети;
5. Выполнить упражнение.

# 3 Выполнение лабораторной работы

1. В своём рабочем каталоге создаём директорию mip, в которой будут выполняться лабораторные работы. Внутри mip создаём директорию lab-ns, а в ней файл shablon.tcl. (рис. 1).

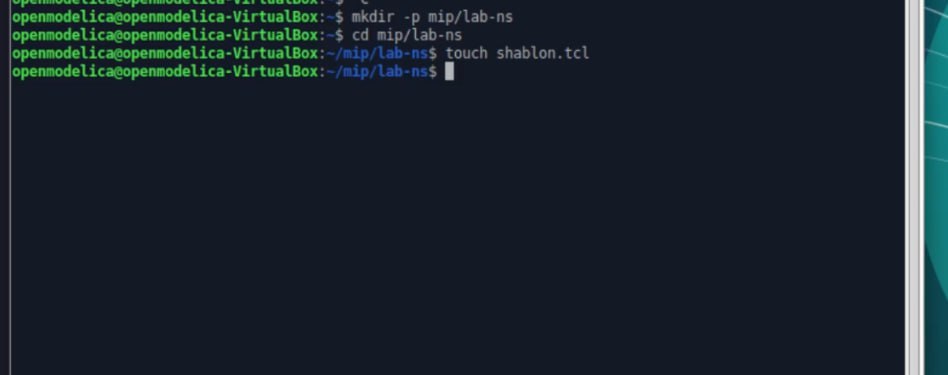


Рис. 1: Создание каталога и файла

1. Открываем на редактирование файл shablon.tcl. Сначала создадим объект типа Simulator, создадим переменную nf и укажем, что требуется открыть на запись nam-файл для регистрации выходных результатов моделирования, вторая строка даёт команду записывать все данные о динамике модели в файл out.nam. Далее создадим переменную f и откроем на запись файл трассировки для регистрации всех событий модели. После этого добавим процедуру finish, которая закрывает файлы трассировки и запускает nam. С помощью команды at указываем планировщику событий, что процедуру finish запустим через 5 с после начала моделирования, после чего запустим симулятор ns. (рис. 2).

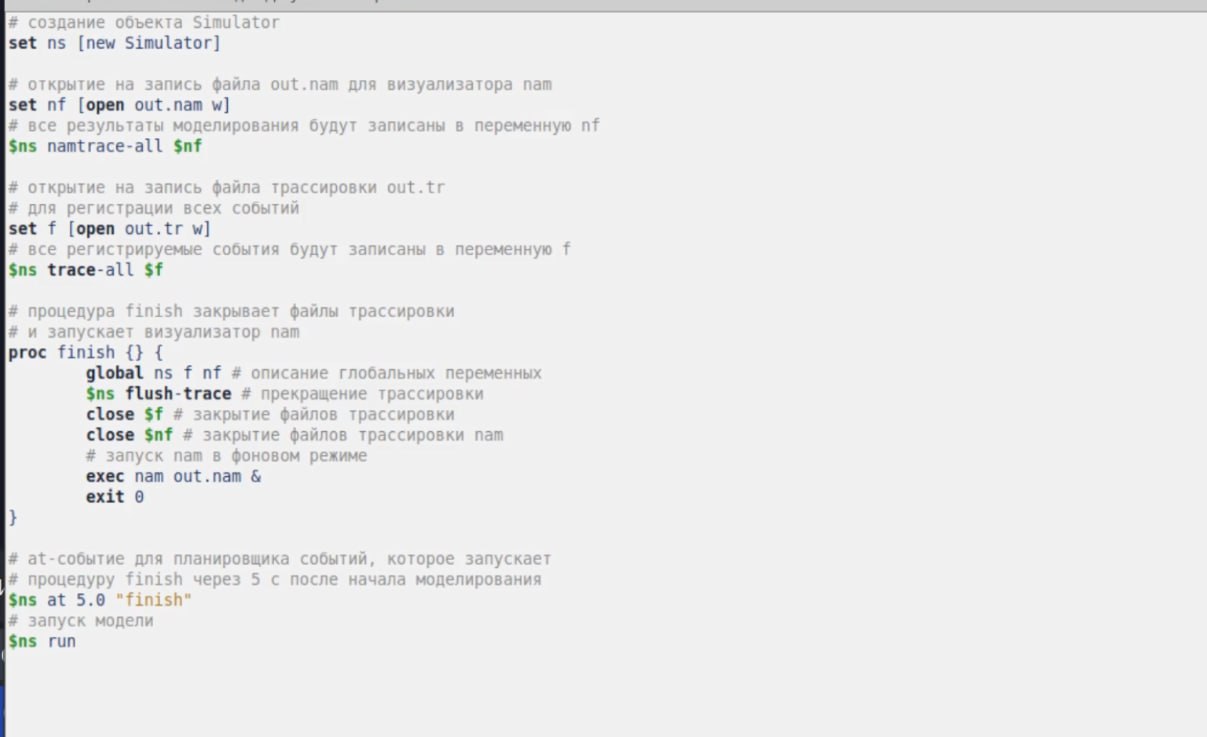


Рис. 2: Редактирование shablon.tcl

1. Сохранив изменения в отредактированном файле shablon.tcl и закрыв его, запустим симулятор командой ns shablon.tcl. Увидим пустую область моделирования, поскольку ещё не определены никакие объекты и действия (рис. 3).

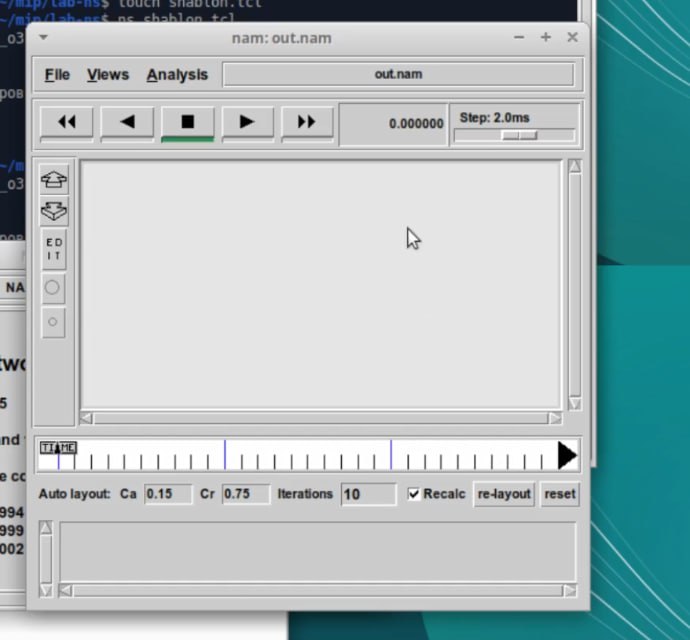


Рис. 3: Пустая область моделирования

1. Выполним второй пример, который посвящён описанию топологии сети, состоящей из двух узлов и одного соединения. Нам 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. (рис. 4).

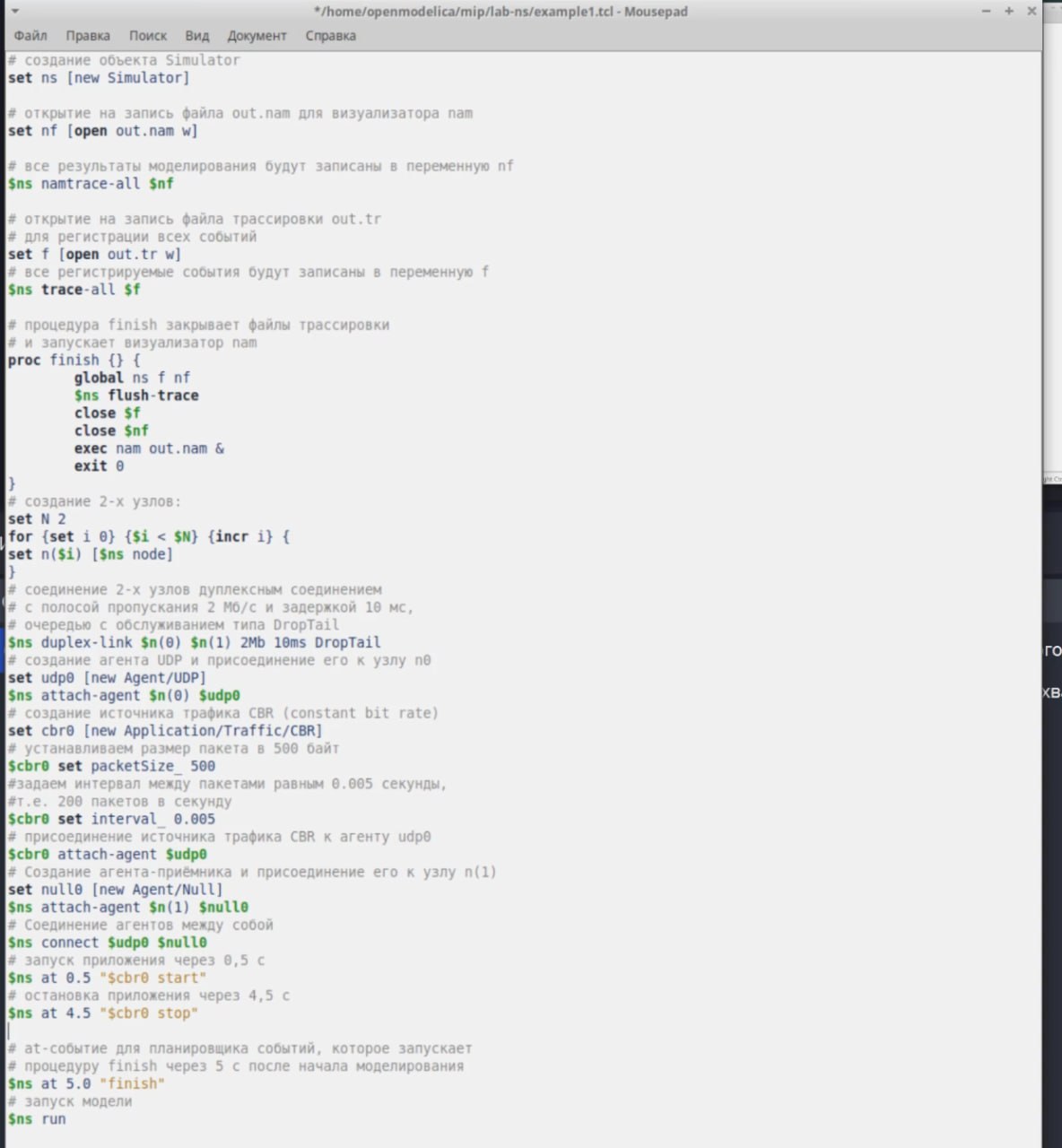


Рис. 4: Отредактированный файл для второго примера

1. Сохранив изменения в отредактированном файле и запустив симулятор, получим анимированный результат моделирования (рис. 5).

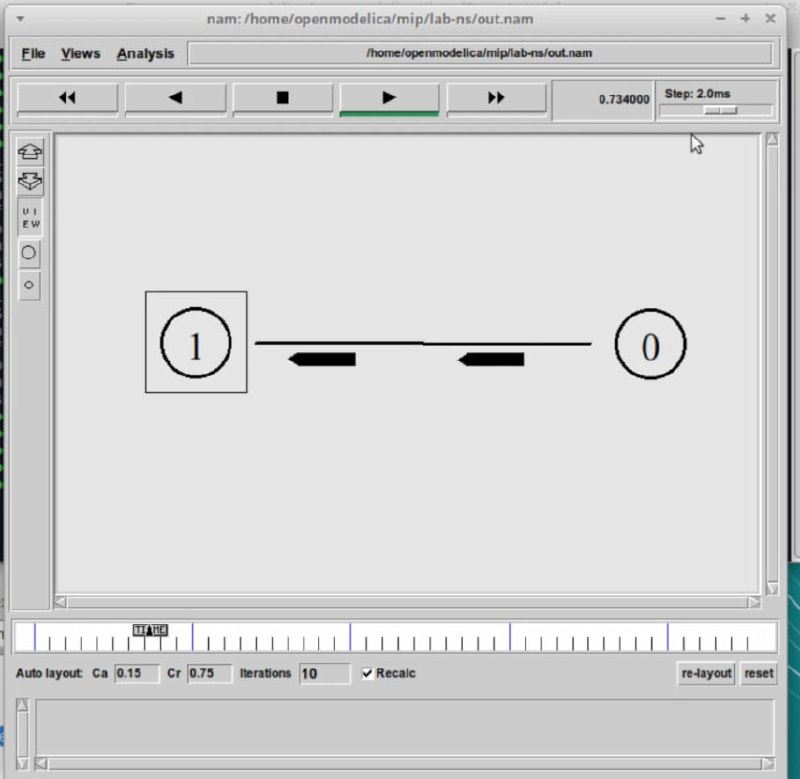


Рис. 5: Второй пример

1. Выполним третий пример. Описание моделируемой сети:

* 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 DropTail 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.

Откроем example2.tcl на редактирование, создадим 4 узла и 3 дуплексных соединения с указанием направления, создадим агент UDP с прикреплённым к нему источником CBR и агент TCP с прикреплённым к нему приложением FTP, создадим агенты-получатели, соединим агенты udp0 и tcp1 и их получателей, зададим описание цвета каждого потока, выполним отслеживание событий в очереди и наложение ограничения на размер очереди, добавим at-события (рис. 6).

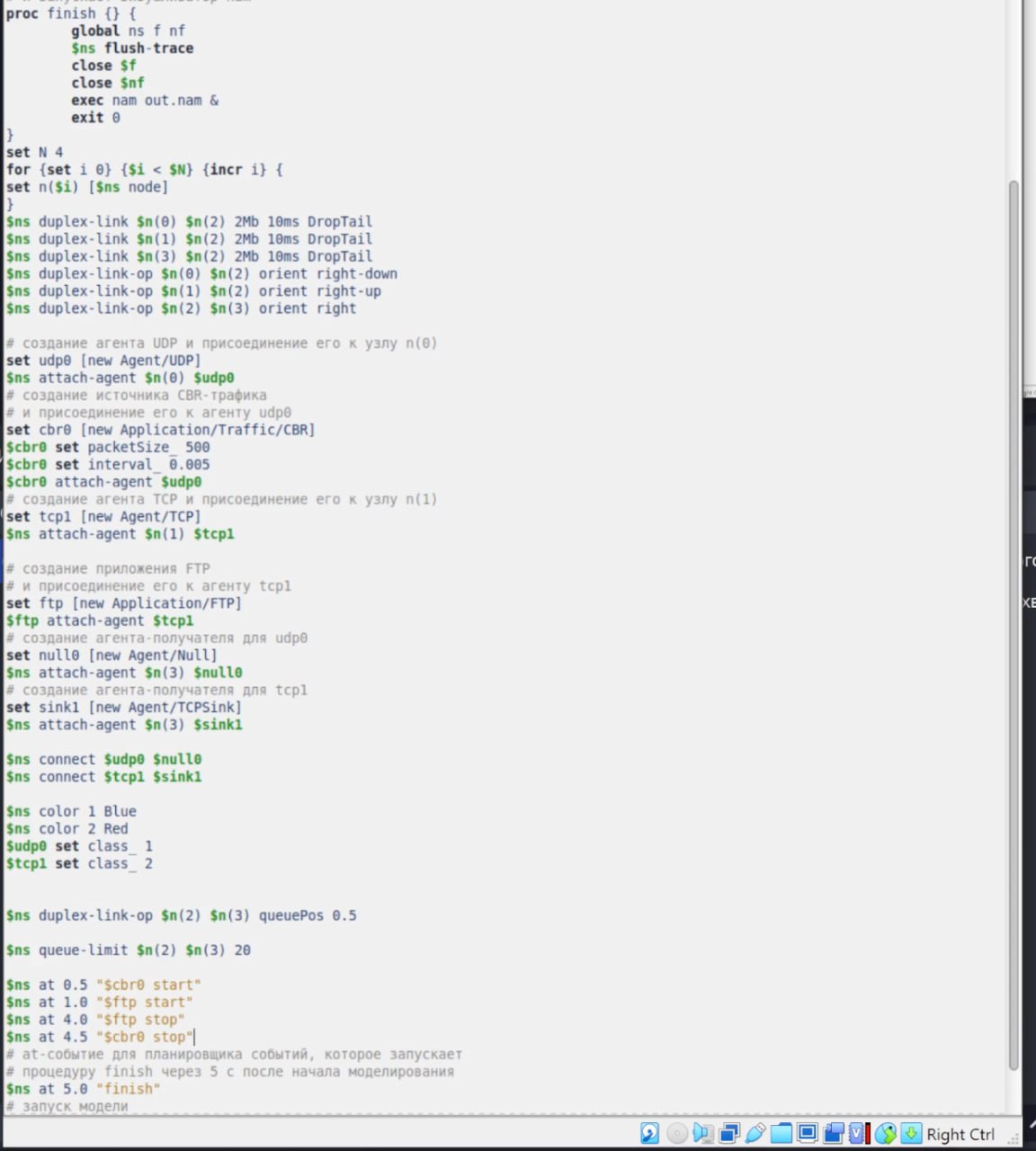


Рис. 6: Третий пример

1. Сохранив изменения в отредактированном файле и запустив симулятор, получим анимированный результат моделирования (рис. 7).

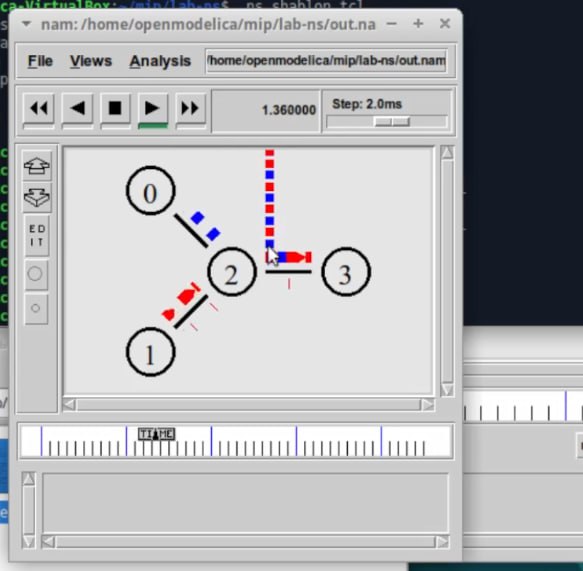


Рис. 7: Результат третьего примера

1. Описание модели 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.

Откроем example3.tcl на редактирование. Опишем топологию моделируемой сети, соединим узлы так, чтобы создать круговую топологию. Каждый узел, за исключением последнего, соединяется со следующим, последний соединяется с первым, для этого в цикле использован оператор %, означающий остаток от деления нацело. Зададим передачу данных от узла n(0) к узлу n(3). Данные передаются по кратчайшему маршруту от узла n(0) к узлу n(3), через узлы n(1) и n(2) (рис. 8).

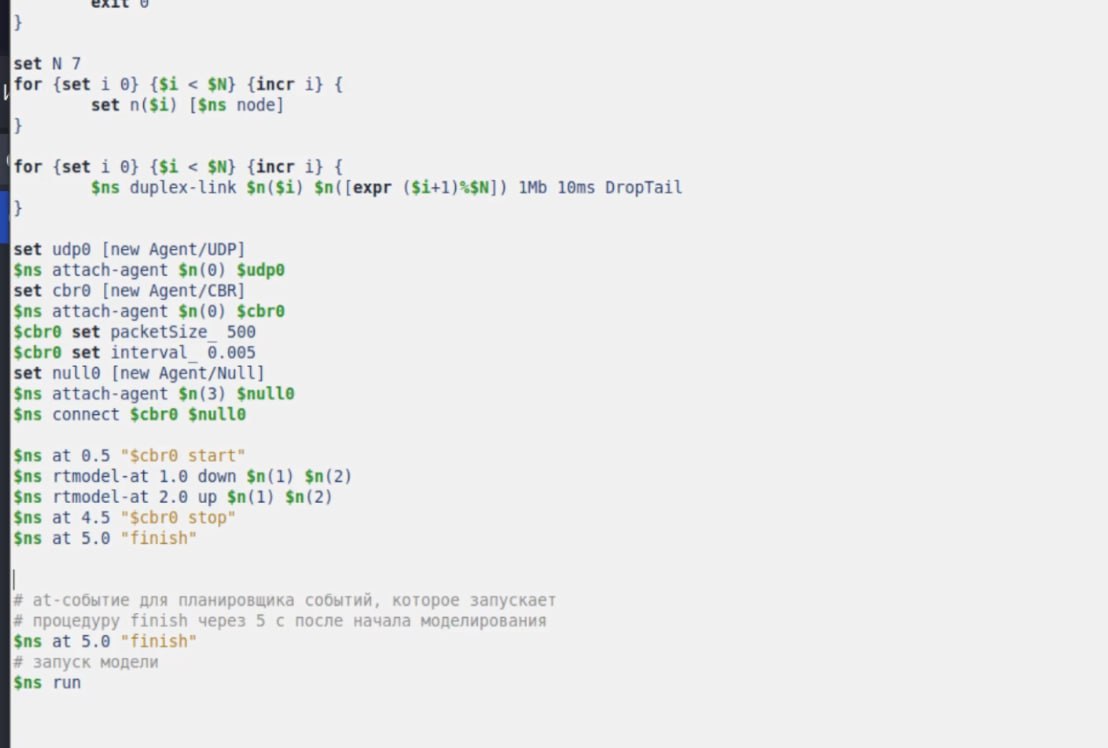


Рис. 8: Четвёртый пример

1. Сохранив изменения в отредактированном файле и запустив симулятор, получим анимированный результат моделирования

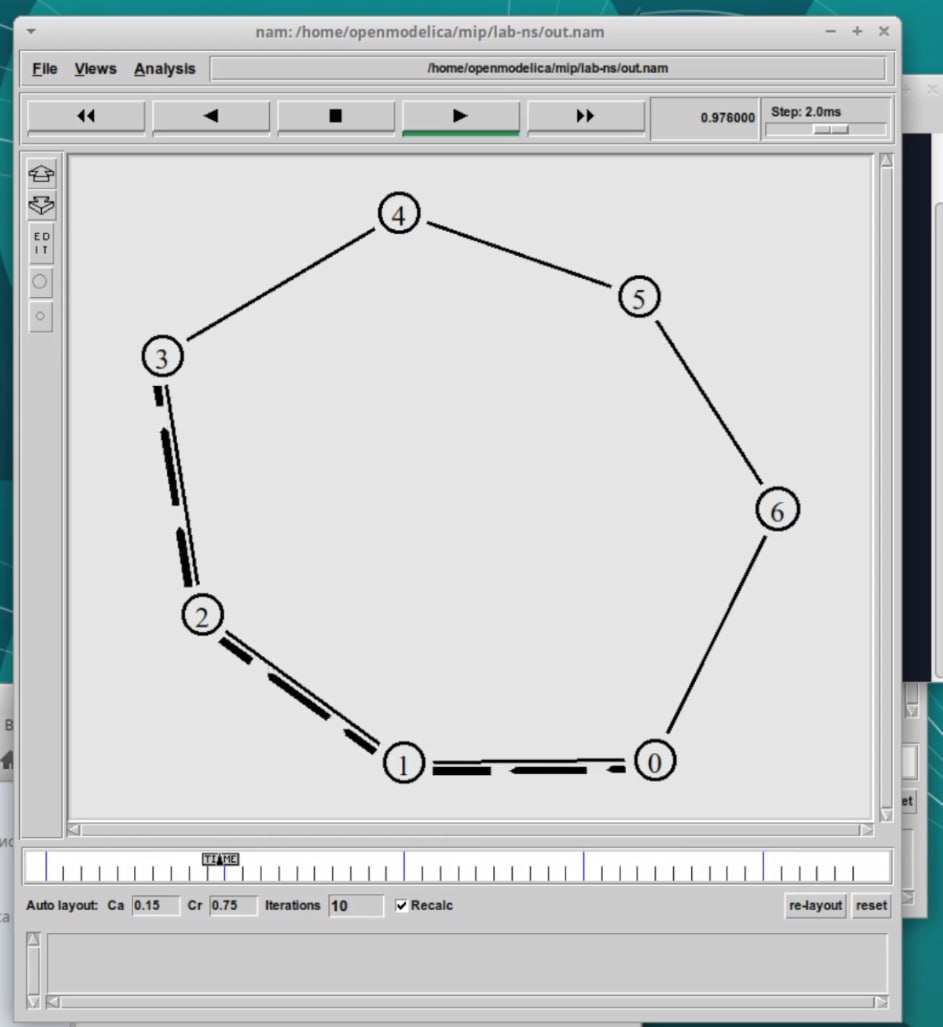


Рис. 9: Результат четвёртого примера

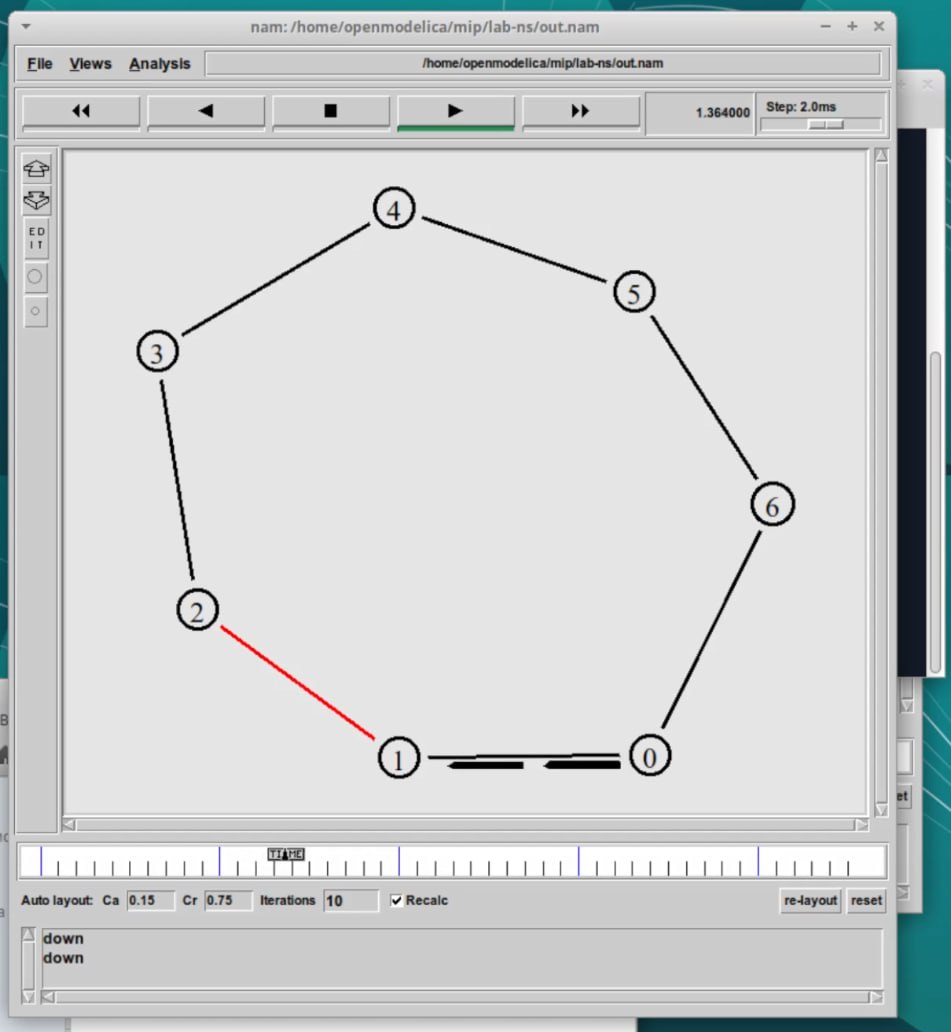


Рис. 10: Результат четвёртого примера

1. Добавив в начало скрипта $ns rtproto DV и увидим, что сразу после запуска в сети отправляется небольшое количество маленьких пакетов, используемых для обмена информацией, необходимой для маршрутизации между узлами, а когда соединение будет разорвано, информация о топологии будет обновлена, пакеты будут отсылаться по новому маршруту через узлы n(6), n(5) и n(4).

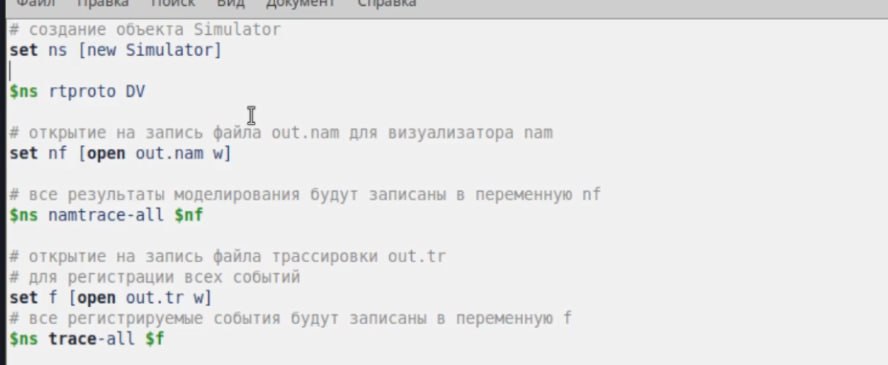


Рис. 11: Добавление строки $ns rtproto DV

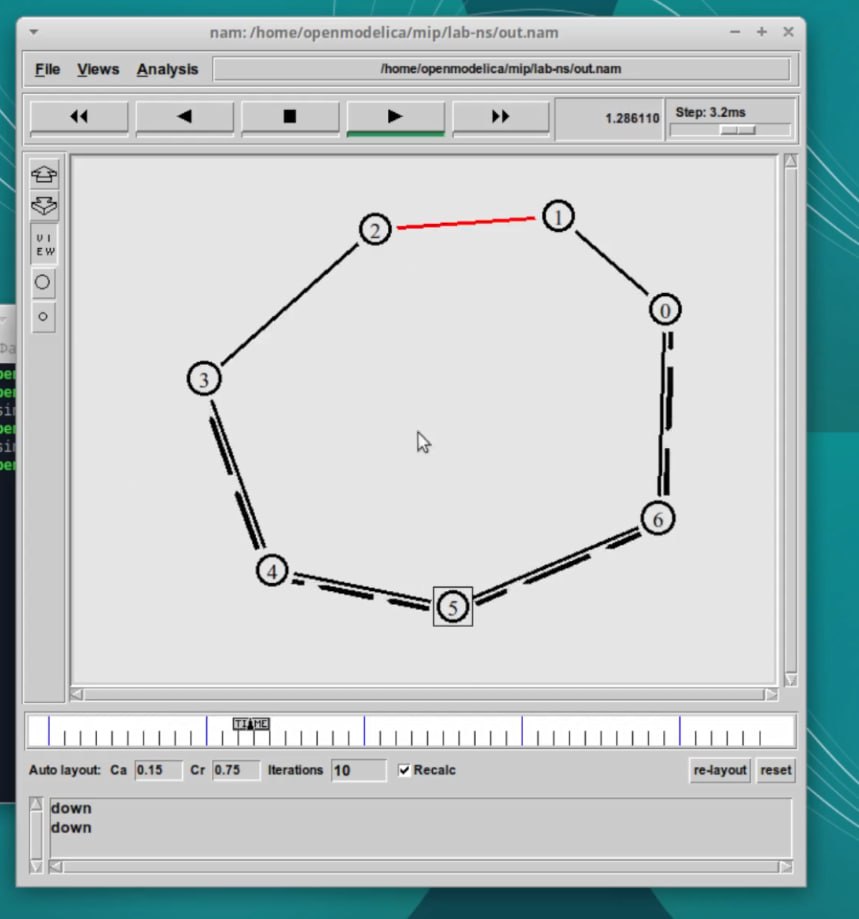


Рис. 12: Результат четвёртого примера после изменений

1. Выполним упражнение, для этого внесем следующие изменения в реализацию примера с кольцевой топологией сети:

* 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 принимающей 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 резервный, после восстановления соединения пакеты снова должны пойти по кратчайшему пути (рис. 13).

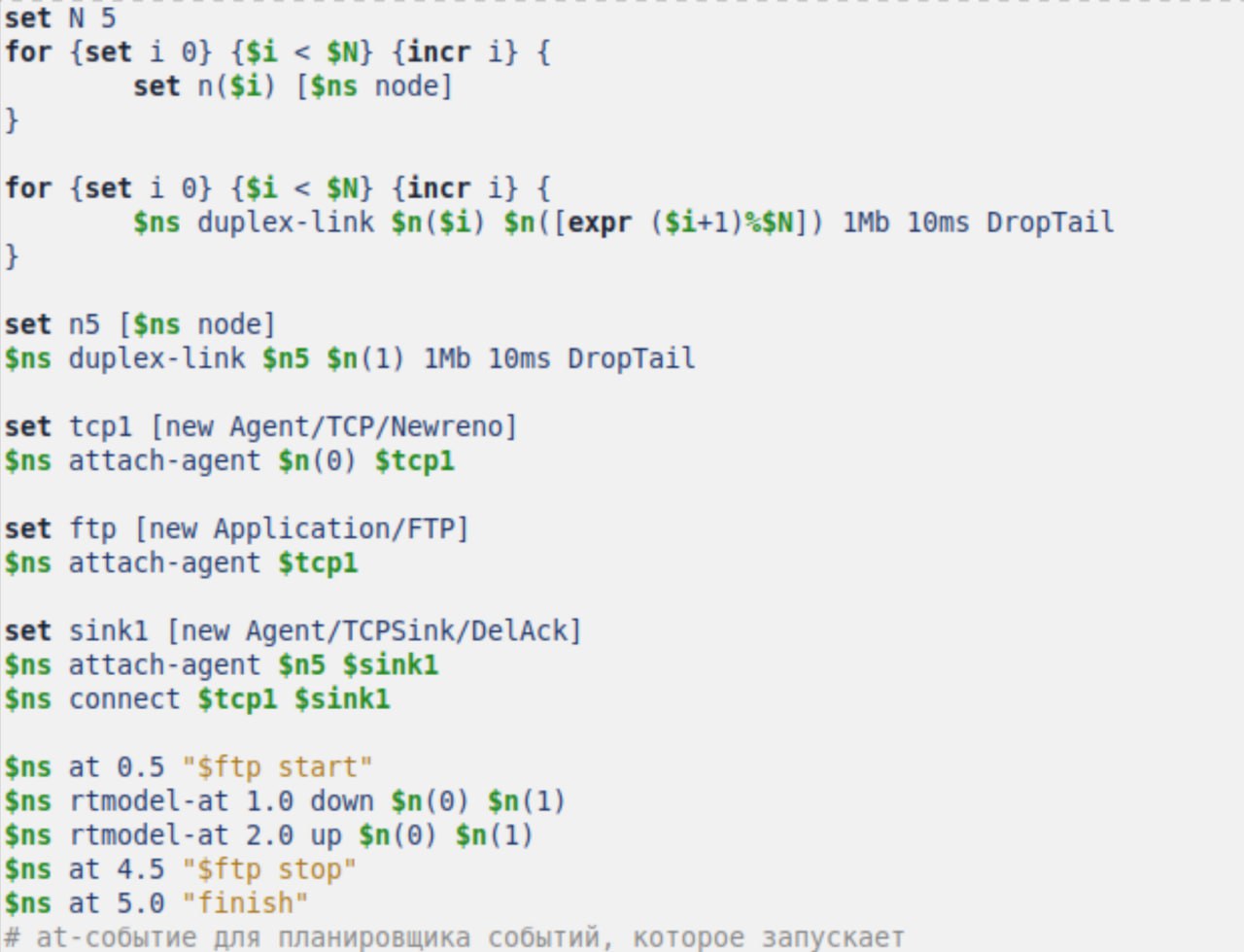


Рис. 13: Упражнение

# 4 Выводы

В процессе выполнения данной лабораторной работы я приобрела навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также проанализировала полученные результаты моделирования.