Лабораторная работа 1

Простые модели компьютерной сети

Туем Гислен

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Приобретить навыков моделирования сетей передачи данных с помощью сред- ства 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

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

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

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.

Скопируем содержимое созданного шаблона в новый файл: cp shablon.tcl example1.tcl и откроем example1.tcl на редактирование. Добавим в него до строки $ns at 5.0 “finish” описание топологии сети. Создадим агенты для генерации и приёма трафика. Создается агент UDP и присоединяется к узлу n0. В узле агент сам не может генерировать трафик, он лишь реализует протоколы и алгоритмы транспортного уровня. Поэтому к агенту присоединяется приложение. В данном случае — это источник с постоянной скоростью (Constant Bit Rate, CBR), который каждые 5 мс посылает пакет R = 500 байт. Таким образом, скорость источника: R = (500⋅8/0.005)=800000 бит/с. Далее создадим Null-агент, который работает как приёмник трафика, и прикрепим его к узлу n1. Соединим агенты между собой. Для запуска и остановки приложения CBR добавляются at-события в планировщик событий (рис. 1).

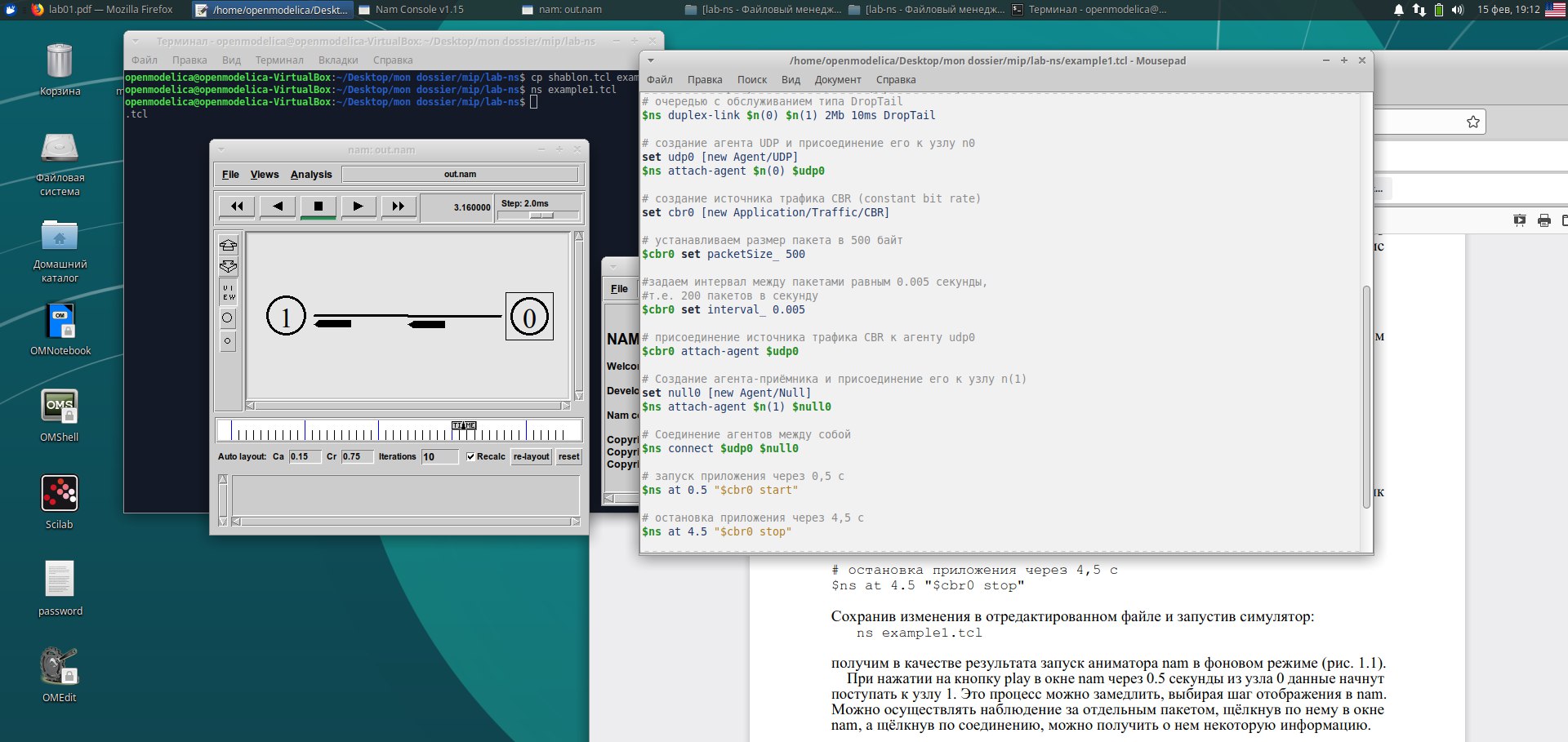


Рис. 1: Визуализация простой модели сети с помощью nam

При нажатии на кнопку play в окне nam через 0.5 секунды из узла 0 данные начнут поступать к узлу 1.

## 3.2 Пример с усложнённой топологией 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.

Скопируем содержимое созданного шаблона в новый файл:cp shablon.tcl example2.tcl и откроем example2.tcl на редактирование. Создадим 4 узла и 3 дуплексных соединения с указанием направления, cоздадим агент UDP с прикреплённым к нему источником CBR и агент TCP с прикреплённым к нему приложением FTP, cоздадим агенты-получатели, cоединим агенты udp0 и tcp1 и их получателей, зададим описание цвета каждого потока, отслеживание событий в очереди, наложение ограничения на размер очереди, добавление at-событий.

Сохранив изменения в отредактированном файле и запустив симулятор, получим анимированный результат моделирования(рис. 2).

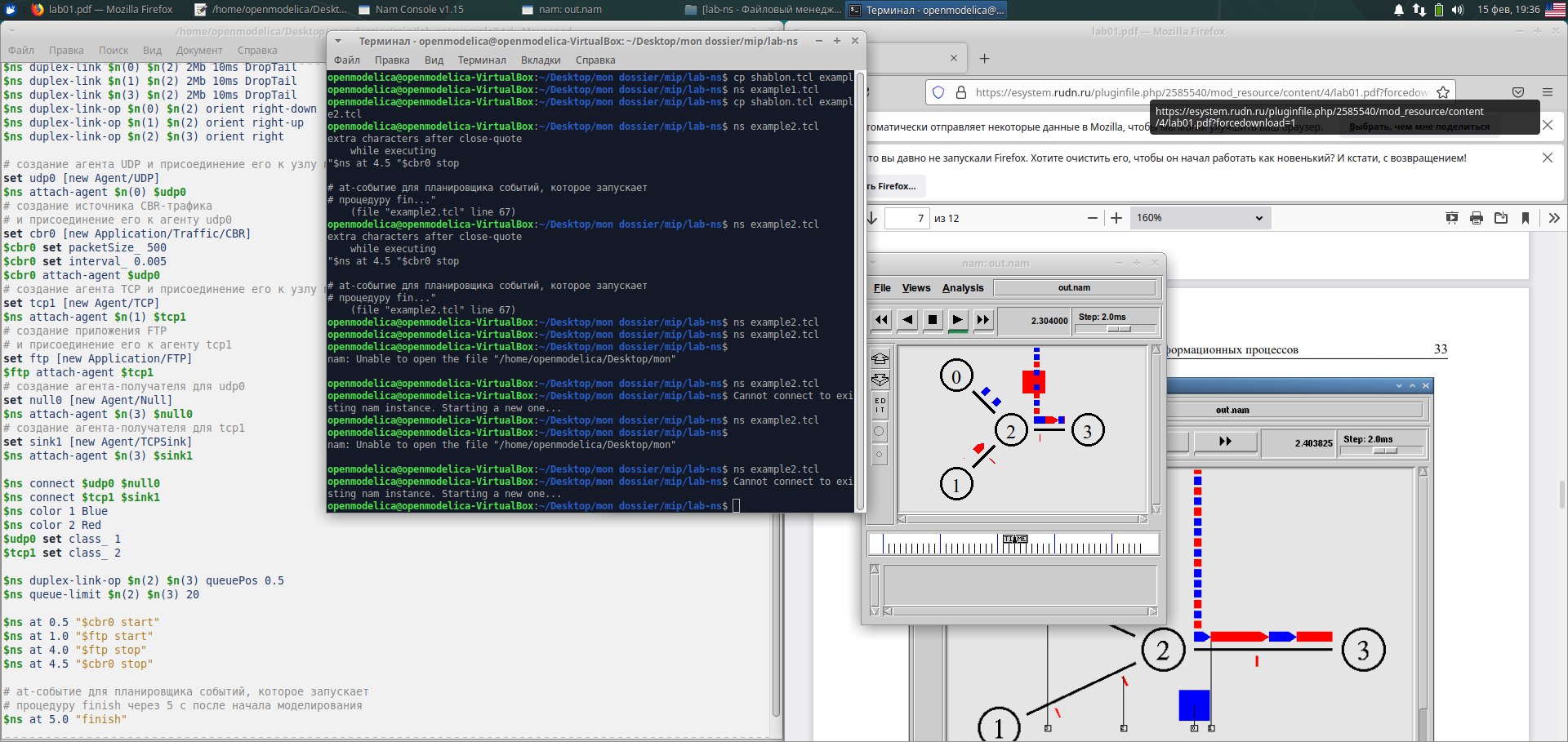


Рис. 2: Мониторинг очереди в визуализаторе nam

## 3.3 Пример с кольцевой топологией сети

Требуется построить модель передачи данных по сети с кольцевой топологией и динамической маршрутизацией пакетов: - сеть состоит из 7 узлов, соединённых в кольцо; - 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 резерв- ный.

Скопируем содержимое созданного шаблона в новый файл: cp shablon.tcl example3.tcl и откроем example3.tcl на редактирование.

Опишем топологию моделируемой сети, далее 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), добавим команду разрыва соединения между узлами n(1) и n(2) на время в одну секунду, а также время начала и окончания передачи данных.

Передача данных при кольцевой топологии сети в случае разрыва соединения представлена(рис. 3).

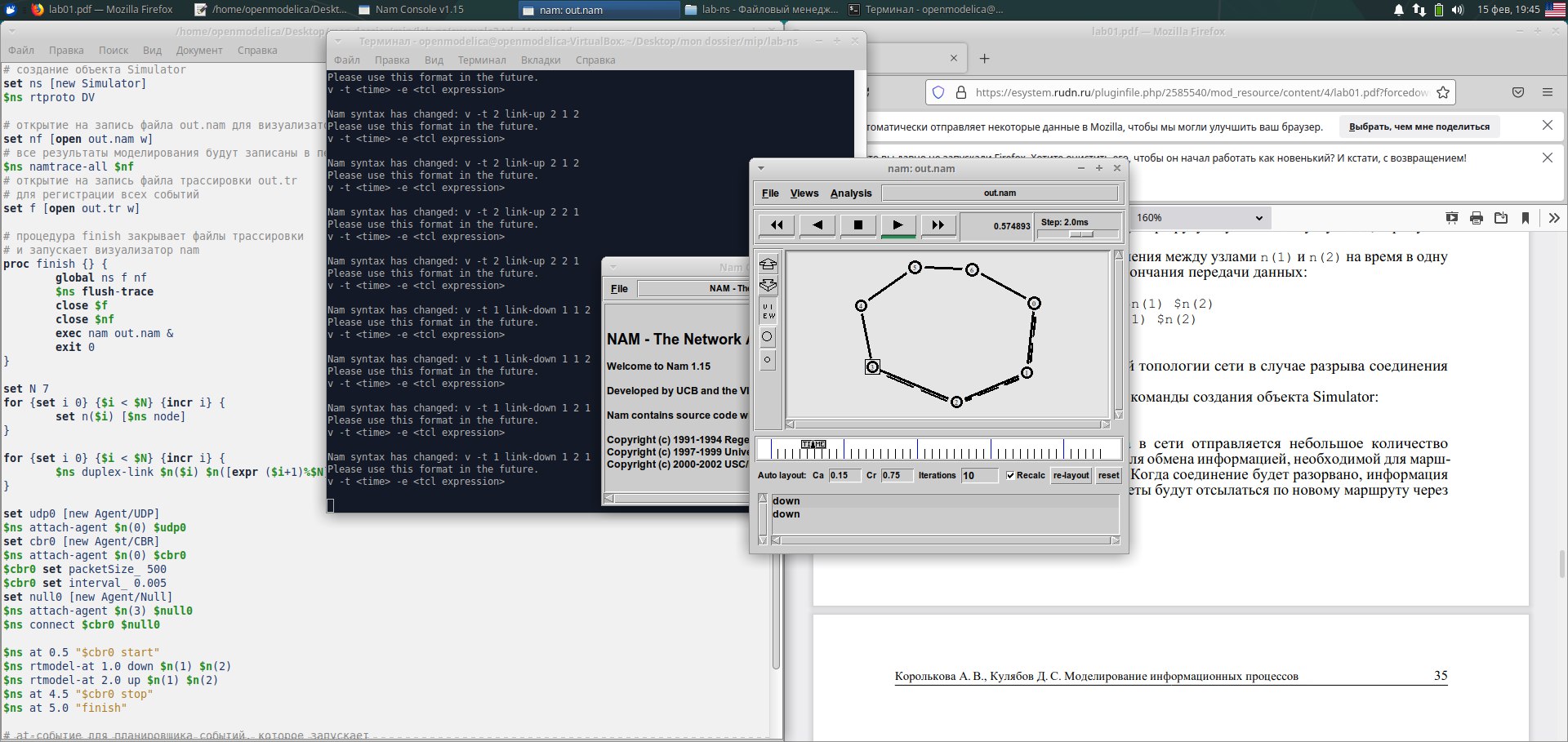


Рис. 3: Передача данных по кратчайшему пути сети с кольцевой топологией

увидим, что сразу после запуска в сети отправляется небольшое количество маленьких пакетов, используемых для обмена информацией, необходимой для маршрутизации между узлами (рис. 1.6). 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)(рис. 4).

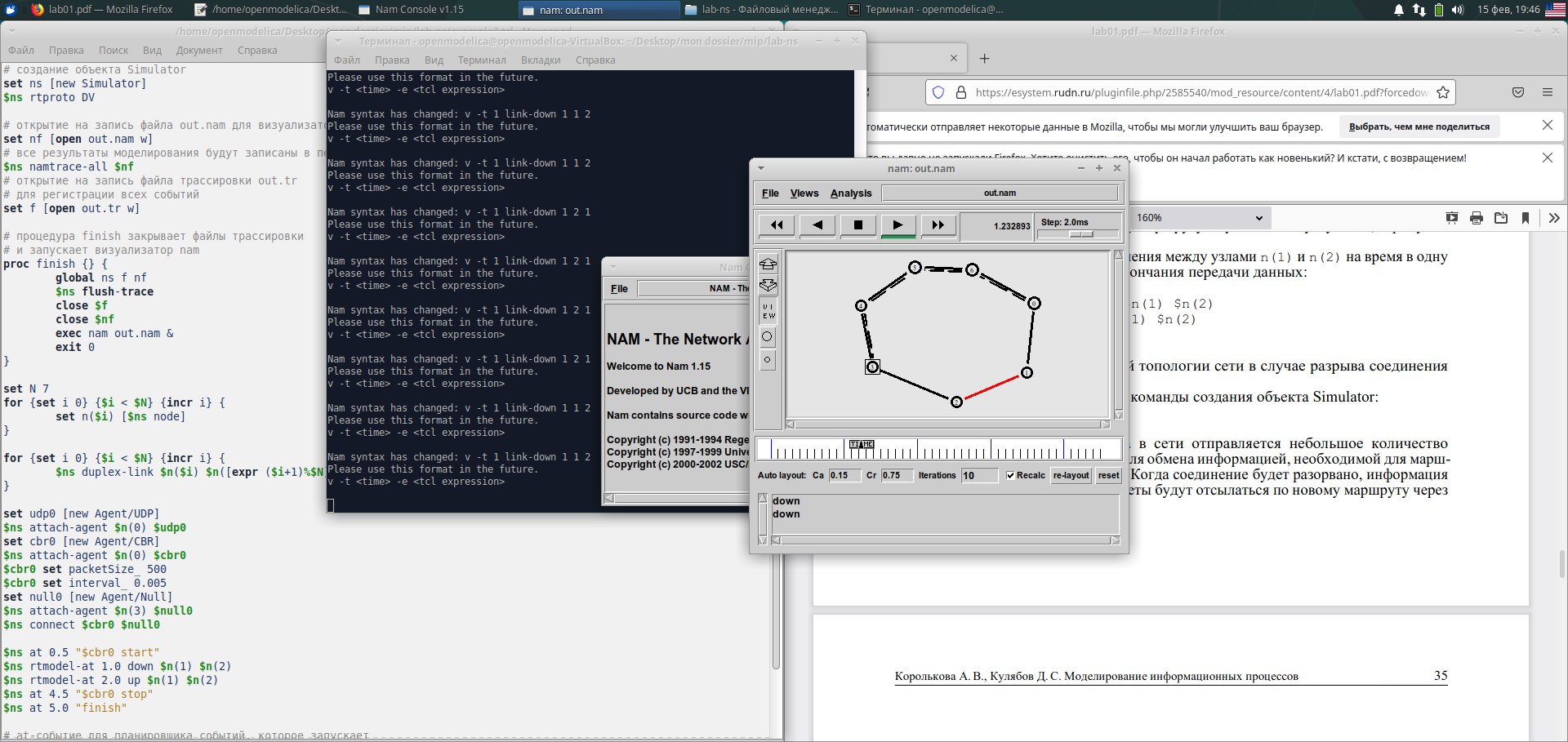


Рис. 4: Маршрутизация данных по сети с кольцевой топологией в случае разрыва соединения

# 4 Упражнение

Внесите следующие изменения в реализацию примера с кольцевой топологией сети: - топология сети должна соответствовать представленной на рис. 1.7; - 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(рис. 5).; - с 1 по 2 секунду модельного времени происходит разрыв соединения между узлами n(0) и n(1)(рис. 6).; - при разрыве соединения маршрут передачи данных должен измениться на резервный, после восстановления соединения пакеты снова должны пойти по кратчайшему пути.

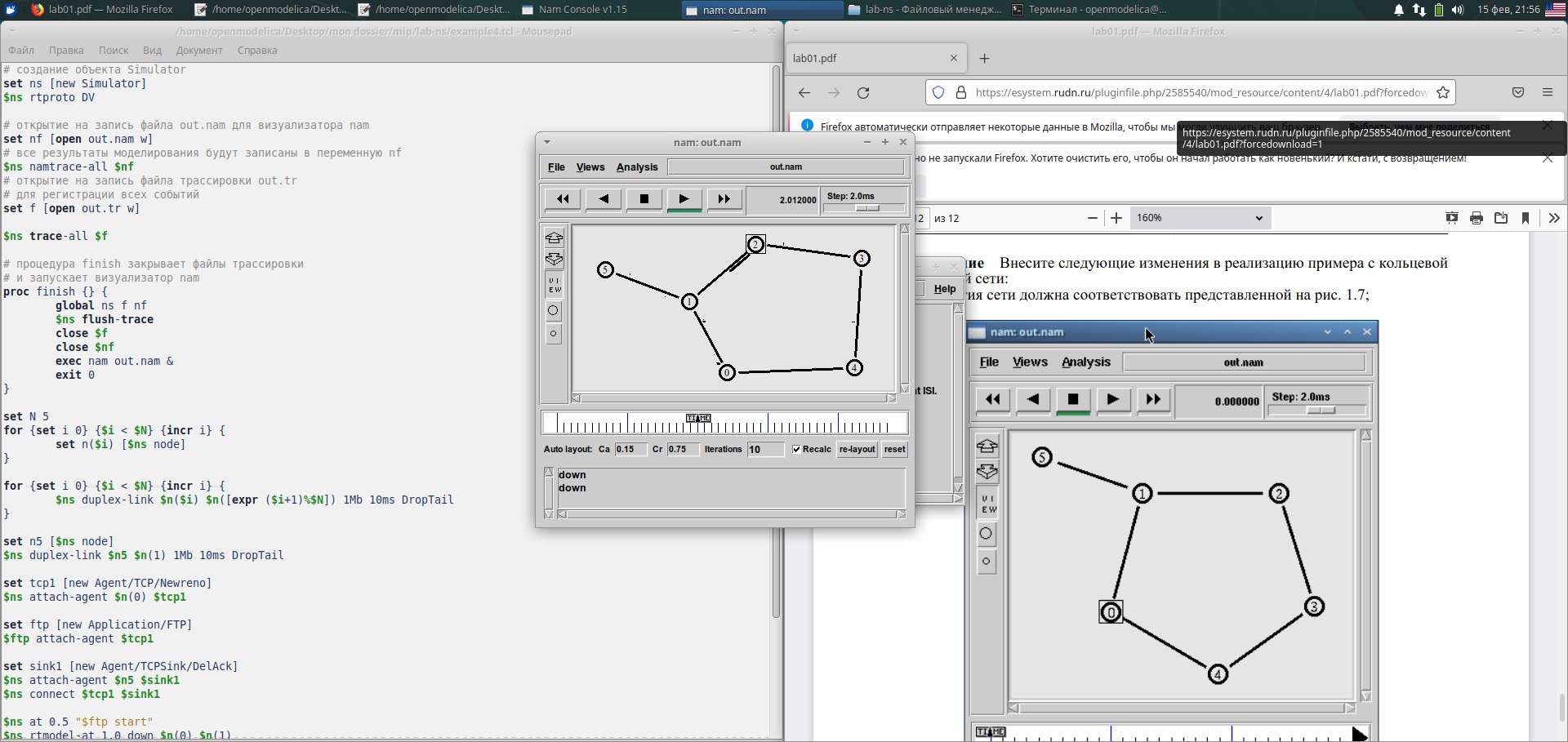


Рис. 5: Изменённая кольцевая топология сети

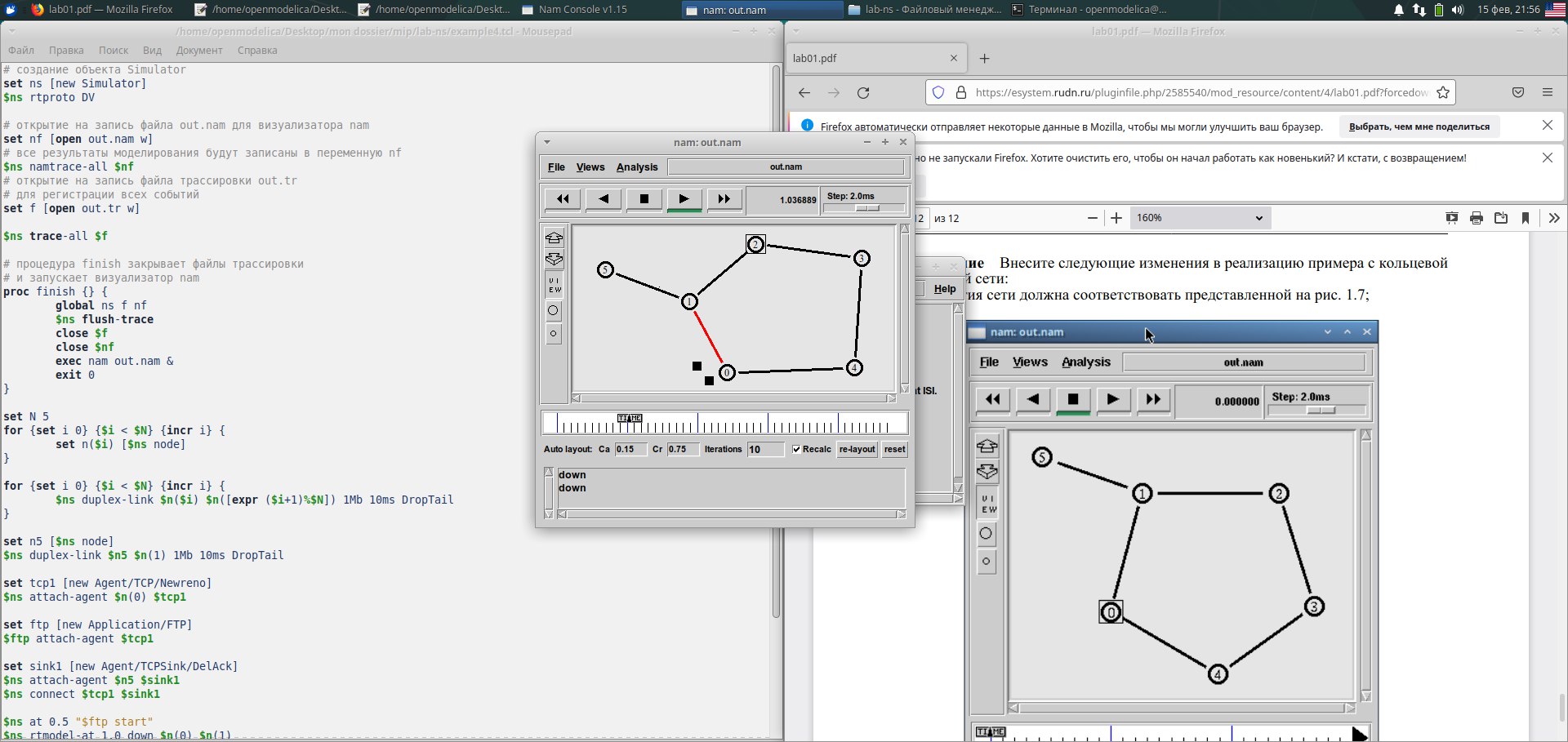


Рис. 6: Разрыв

# 5 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.