

# **Лабораторная работа 1**

**Имитационное моделирование**

Голощапов Ярослав Вячеславович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>13</b>

## Список иллюстраций

4.1	Настройка шаблона . . . . .	8
4.2	Реализация модели . . . . .	9
4.3	Реализация усложненной модели . . . . .	10
4.4	Пример с кольцевой топологией . . . . .	10
4.5	Передача данных по сети с кольцевой топологией в случае разрыва соединения . . . . .	11
4.6	Изменённая кольцевая топология сети . . . . .	12
4.7	Изменённая кольцевая топология сети в случае разрыва соединения	12

## **Список таблиц**

# 1 Цель работы

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

## **2 Задание**

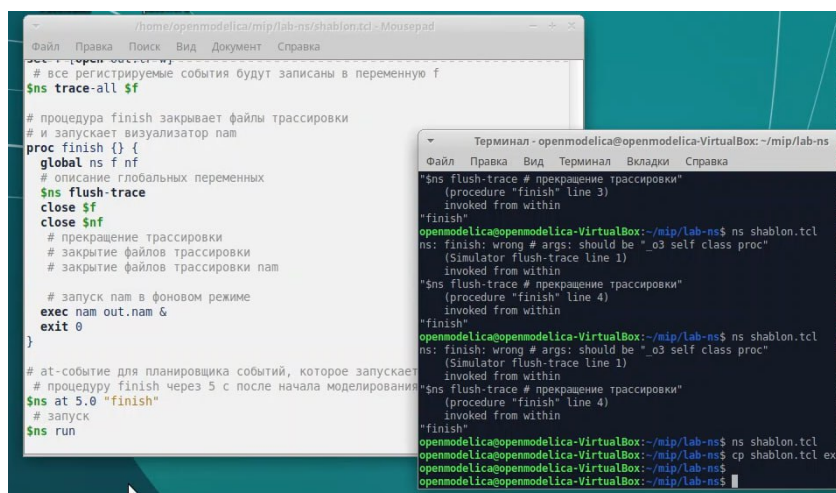
Получить практические навыки на примерах и выполнить упражнение

## 3 Теоретическое введение

Network Simulator (NS-2) — один из программных симуляторов моделирования процессов в компьютерных сетях. NS-2 позволяет описать топологию сети, конфигурацию источников и приёмников трафика, параметры соединений (полосу пропускания, задержку, вероятность потерь пакетов и т.д.) и множество других параметров моделируемой системы. Данные о динамике трафика, состоянии соединений и объектов сети, а также информация о работе протоколов фиксируются в генерируемом trace-файле.

## 4 Выполнение лабораторной работы

Для начала было необходимо настроить шаблон для сценария NS-2 (рис. 4.1).



```
File Edit View Document Help
$ns trace-all $f
# процедура finish закрывает файлы трассировки
# и запускает визуализатор nam
proc finish {} {
    global ns f nf
    # описание глобальных переменных
    $ns flush-trace
    close $f
    close $nf
    # прекращение трассировки
    # закрытие файлов трассировки
    # закрытие файлов трассировки nam

    # запуск nam в фоновом режиме
    exec nam out.nam &
    exit 0
}

# at-событие для планировщика событий, которое запускает
# процедуру finish через 5 с после начала моделирования
$ns at 5.0 "finish"
# запуск
$ns run
```

```
Терминал - openmodelica@openmodelica-VirtualBox: ~/mip/lab-ns
$ns flush-trace # прекращение трассировки
(procedure "finish" line 3)
invoked from within
"finish"
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns shablon.tcl
ns: finish: wrong # args: should be "_o3 self class proc"
(Simulator flush-trace line 1)
invoked from within
"$ns flush-trace # прекращение трассировки"
(procedure "finish" line 4)
invoked from within
"finish"
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns shablon.tcl
ns: finish: wrong # args: should be "_o3 self class proc"
(Simulator flush-trace line 1)
invoked from within
"$ns flush-trace # прекращение трассировки"
(procedure "finish" line 4)
invoked from within
"finish"
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns shablon.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp shablon.tcl exi
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$
```

Рис. 4.1: Настройка шаблона

Далее выполнял простой пример описания топологии сети, состоящей из двух узлов и одного соединения. Задача: 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. 4.2).



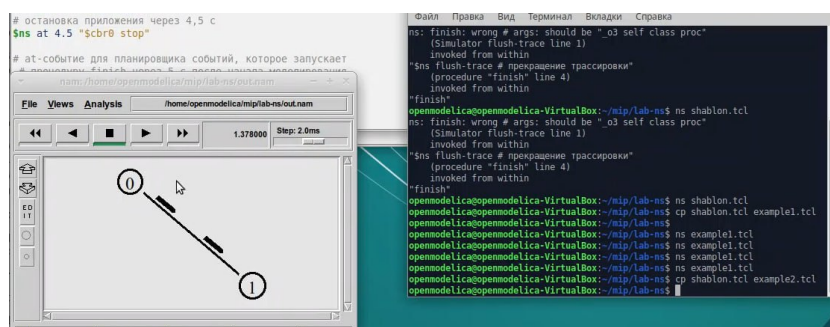


Рис. 4.2: Реализация модели

Пример с усложненной топологией сети.

Задача: 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.(рис. 4.3).

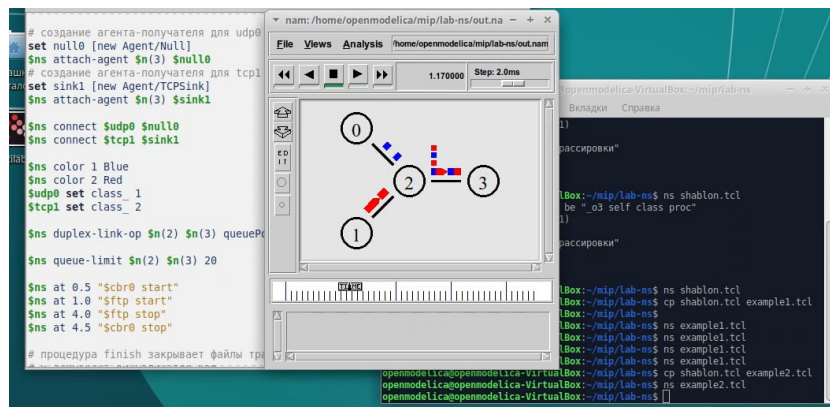


Рис. 4.3: Реализация усложненной модели

Следующая задача: 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. (рис. 4.4). (рис. 4.5)

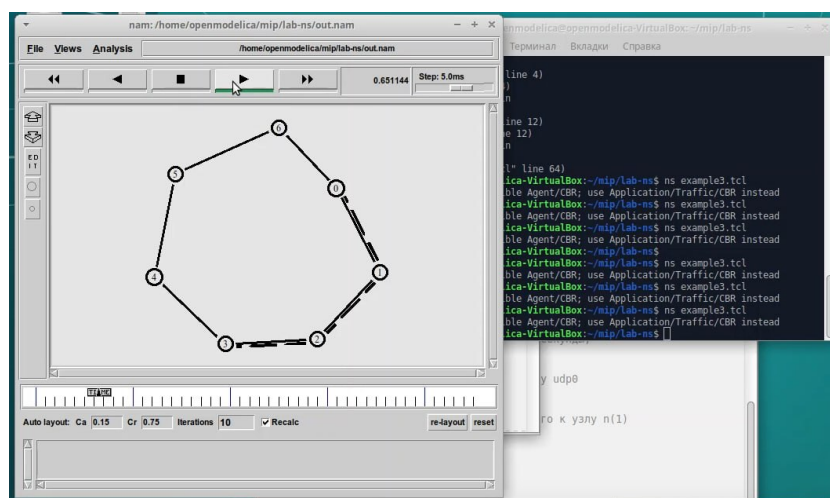


Рис. 4.4: Пример с кольцевой топологией

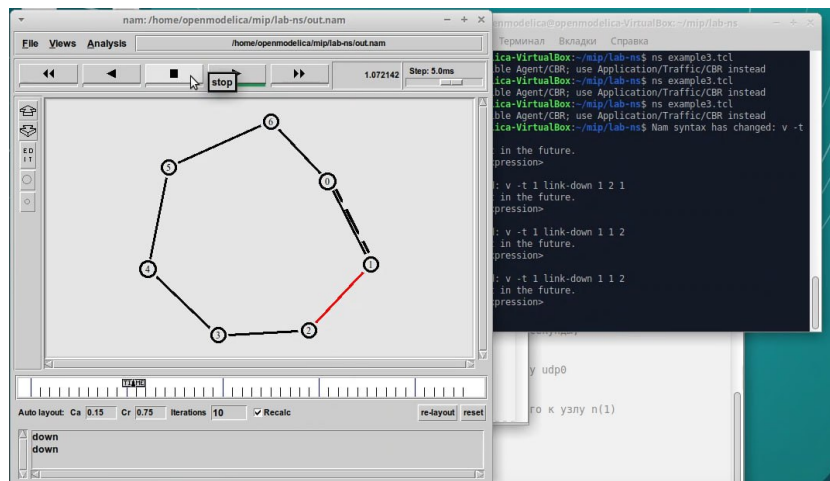


Рис. 4.5: Передача данных по сети с кольцевой топологией в случае разрыва соединения

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; – 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. 4.6) (рис. 4.7)

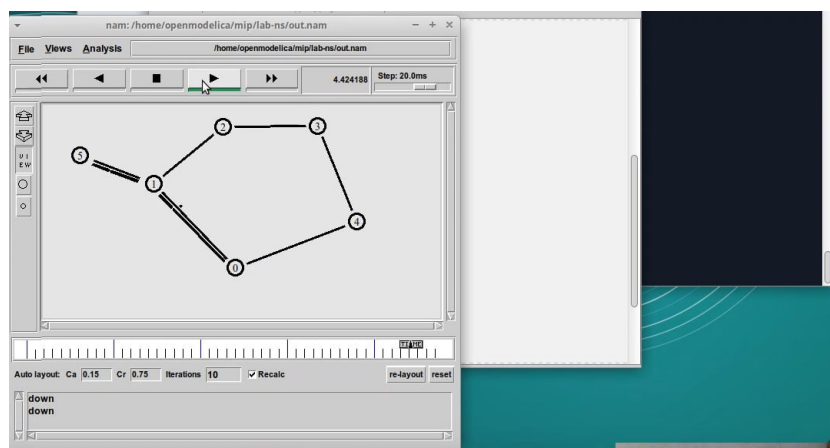


Рис. 4.6: Изменённая кольцевая топология сети

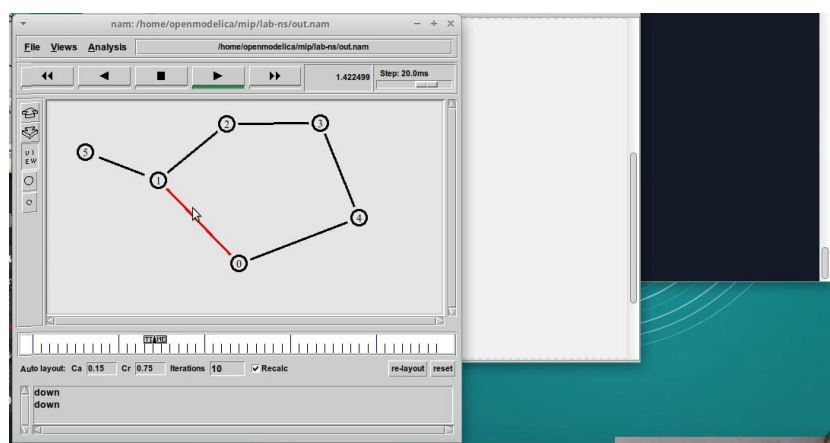


Рис. 4.7: Изменённая кольцевая топология сети в случае разрыва соединения

## **5 Выводы**

В этой лабораторной работе я приобрел навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также проанализировал полученные результаты моделирования