РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

ОТЧЕТ

по лабораторной работе № 1

# дисциплина: Моделирование информационных процессов

Студент: Тейшейра Боа Морте Селмилтон

Группа:НкНбд-01-20

**МОСКВА**

2023г.

# Постановка задачи

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

**Выполнения**

1.2 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.

1.3 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;

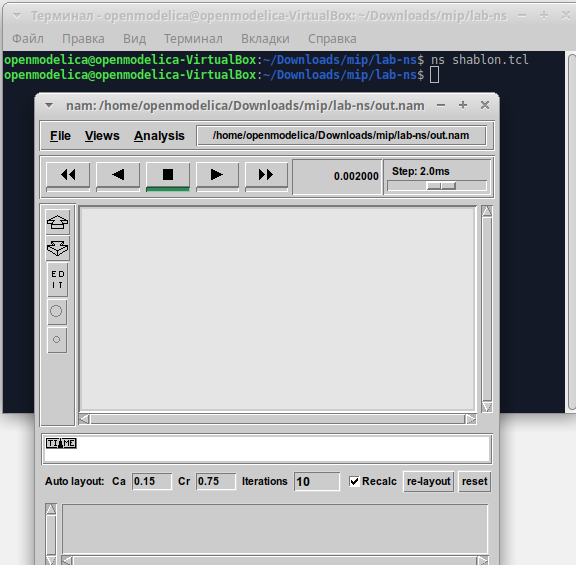
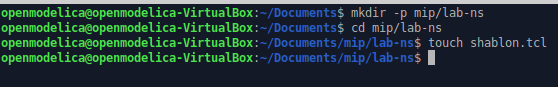
– генератор cbr генерирует пакеты размером 1 Кбайт со скоростью 1 Мбит/с;

– 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.

1.4 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);

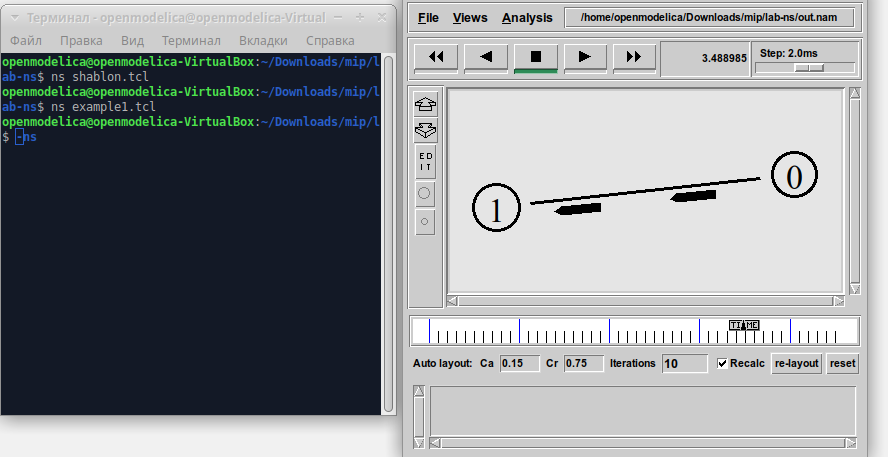
– при разрыве соединения маршрут передачи данных должен измениться на резервный. Упражнение Внесите следующие изменения в реализацию примера с кольцевой топологией сети. Выполнение работы

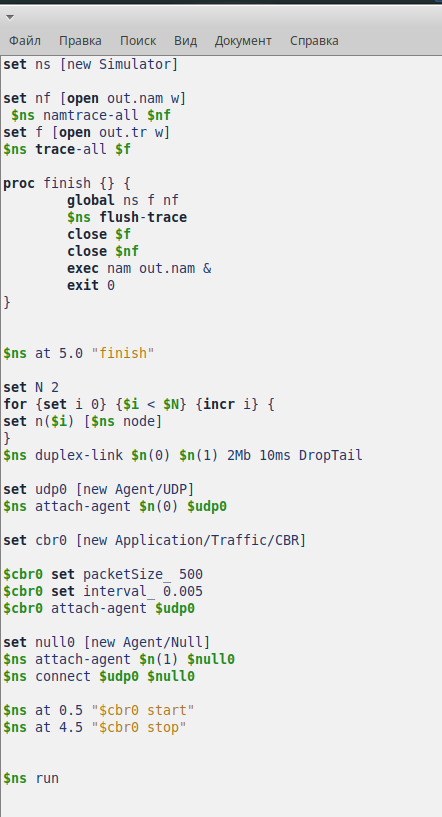
* 1. **Шаблон сценария для NS-2**



в этом примере мы создали шаблон для создания и настройки узлов. наши основные значения для подключения и отсоединения между нашими узлами показаны в виде глобальных переменных.

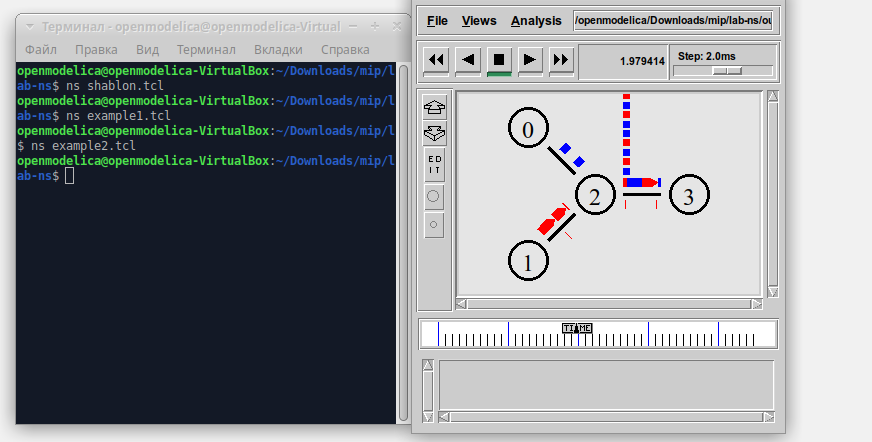
* 1. **Простой пример описания топологии сети, состоящей из двух узлов и одного соединения**

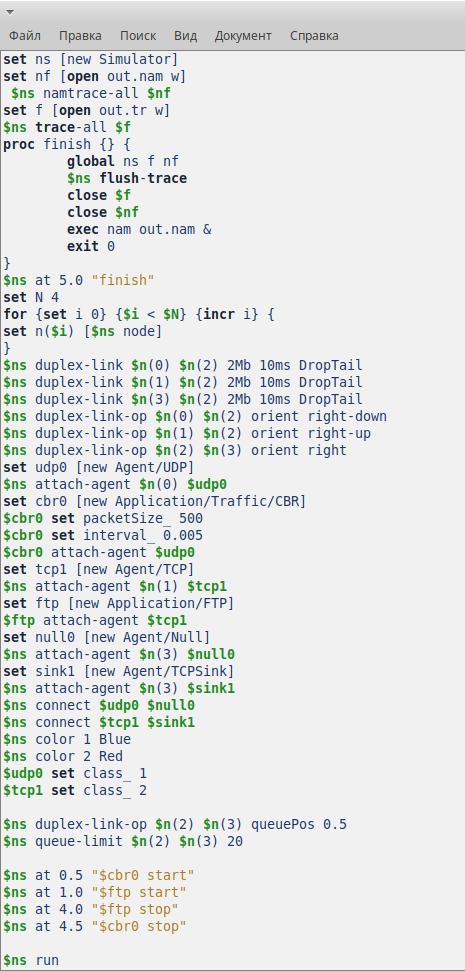




в этом примере мы создали 2 узла, которые подключены вместе, чтобы измерить пропускную способность нашего протокола udp, которая составляет 2 МБ 10 мс. наш протокол udp подключен к cbr для визуализации передачи данных между узлами. 1.3. Пример с усложнённой топологией сети.

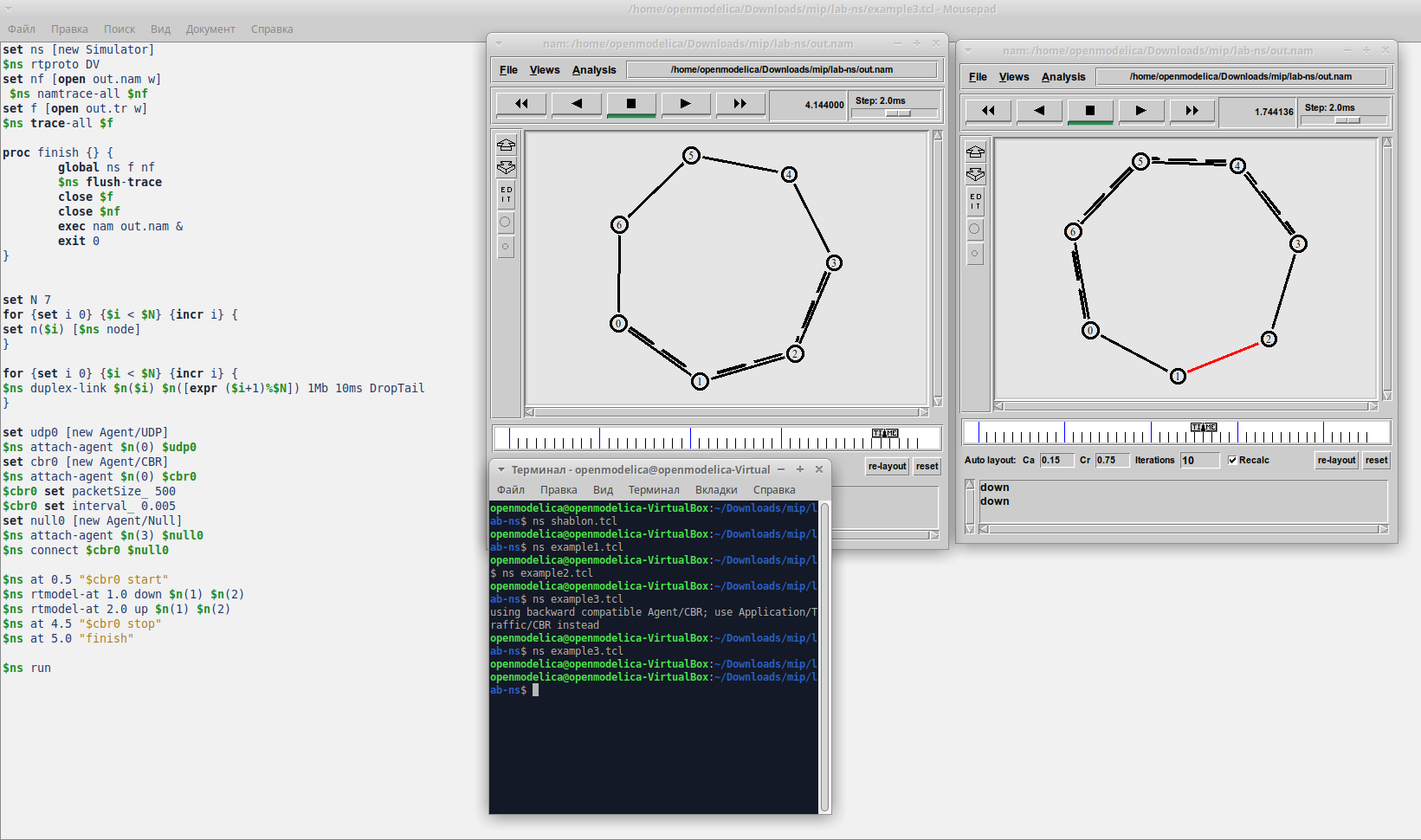
* 1. **Пример с усложнённой топологией сети**





в этом примере мы используем 2 разных протокола, которые являются udp и tcp, и измеряем связь и пропускную способность между ними в заданных определенных часовых поясах, которые для tcp от начала до конца и для udp от 0,5 секунды до 4,5 секунды. мы настроили 4 разных узла, протокол tcp начинается с 1 и заканчивается на 4-м узле, а udp начинается с 0 и также заканчивается на 4. эти полосы пропускания заданы классами и цветом для лучшей проекции.

* 1. **Пример с кольцевой топологией сети**

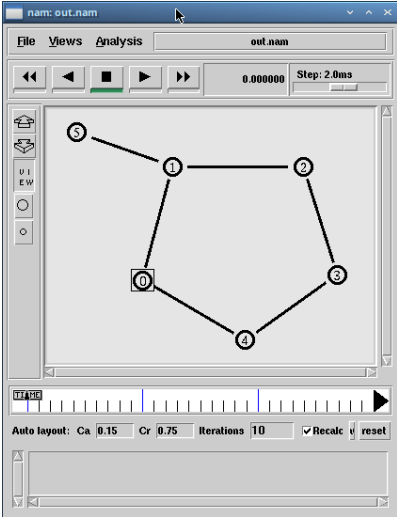
****

в 1.4 мы создаем набор узлов, которые связаны друг с другом в виде круга, начиная с узла 0 в udp и тестируя путь с помощью команды rtremote для запуска up down или down up, чтобы мы могли визуализировать разницу между ссылкой на разных путях.

**Упражнение**

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 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.

**Решение**

в этом примере я создал круг и узел node снаружи, который соединяется с 1-м узлом и переходит с 4-го точно так же, как показано в примере, он начинается с 0 и должен быть завершен на 5-м узле. требования показали, что нам нужно связать его только с помощью tcp, он просто находит кратчайший способ добраться до него по обоим протоколам.

**set ns [new Simulator]**

**$ns rtproto DV**

**set nf [open out.nam w]**

**$ns namtrace-all $nf**

**set f [open out.tr w]**

**$ns trace-all $f**

**proc finish {} {**

**global ns f nf**

**$ns flush-trace**

**close $f**

**close $nf**

**exec nam out.nam &**

**exit 0**

**}**

**set N 6**

**for {set i 0} {$i < $N} {incr i} {**

**set n($i) [$ns node]**

**}**

**for {set i 0} {$i < [expr $N -1 ] } {incr i} {**

**$ns duplex-link $n($i) $n([expr ($i+1)%[expr $N -1 ]]) 1Mb 10ms DropTail**

**}**

**$ns duplex-link $n(5) $n(1) 1Mb 10ms DropTail**

**set tcp0 [new Agent/TCP/Newreno]**

**$ns attach-agent $n(0) $tcp0**

**set sink [new Agent/TCPSink/DelAck]**

**$ns attach-agent $n(5) $sink**

**$ns connect $tcp0 $sink**

**set ftp0 [new Application/FTP]**

**$ftp0 attach-agent $tcp0**

**$ftp0 set type\_ FTP**

**$ns at 0.5 "$ftp0 start"**

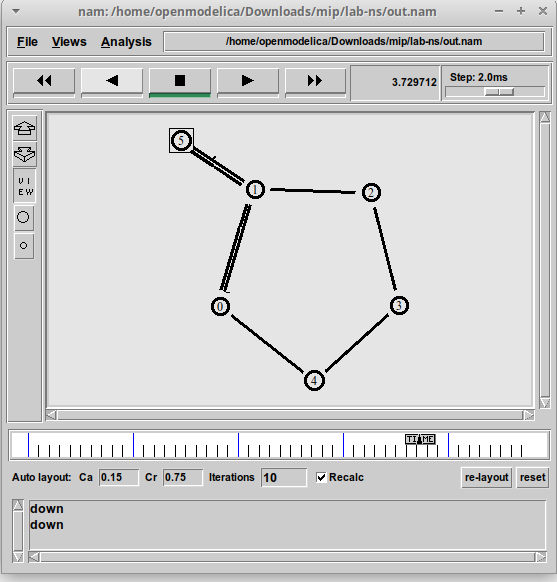
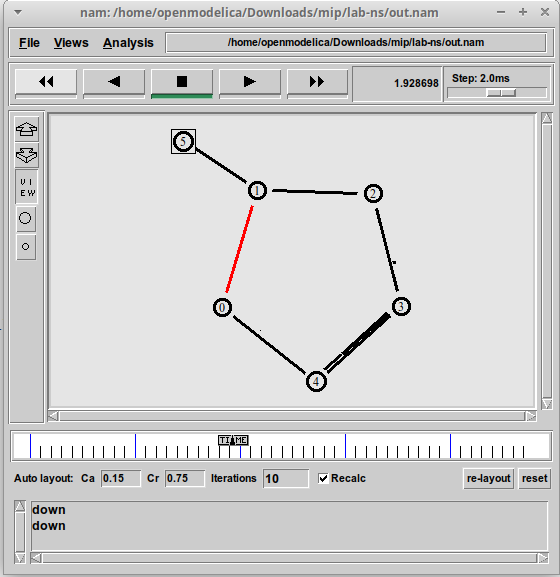
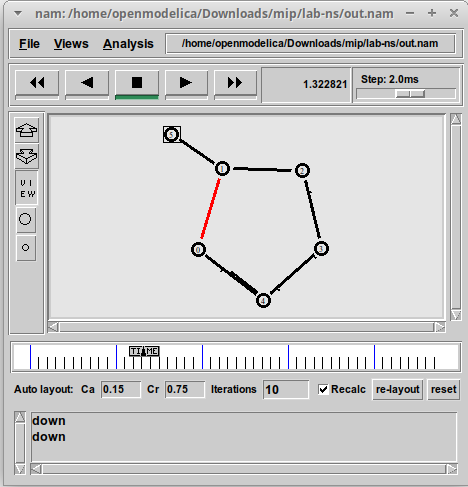
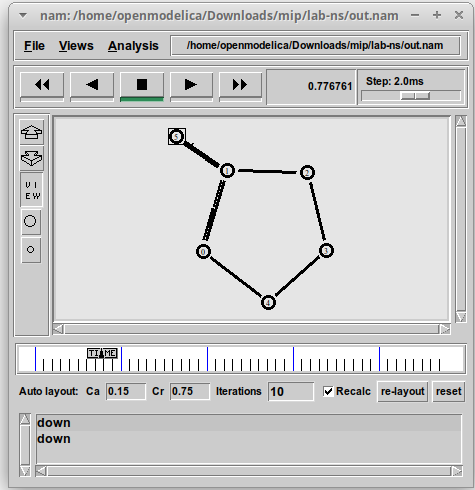
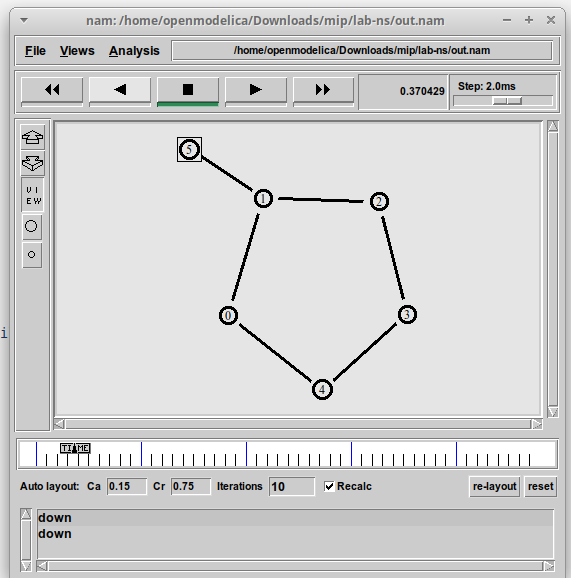
**$ns rtmodel-at 1.0 down $n(0) $n(1)**

**$ns rtmodel-at 2.0 up $n(0) $n(1)**

**$ns at 4.5 "$ftp0 stop"**

**$ns at 5.0 "finish"**

**$ns run**

****

**Заключение**

в лабораторной работе мы изучили программу nc (network connector), выполнив несколько примеров для создания базового шаблона для настройки и подключения узлов, создали базовые примеры 2 узлов, подключив их по протоколу udp, использовали оба протокола udp tcp для визуализации соединения на конечном узле и разницы между ними, создал круг узлов, чтобы визуализировать и научиться связывать их сверху или снизу. в конце мы создали узел Кольца точно так же, как в предыдущем примере, и подключили последний узел извне. мы использовали протокол tcp, чтобы найти кратчайший путь от начала до конца