# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук Кафедра теории вероятностей и кибербезопасности**

# ОТЧЕТ

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

*дисциплина: Моделирование информационных процессов*

Студент: Маслова Анастасия

Группа: НКНбд-01-21

**МОСКВА**

2024 г

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

Постановка задачи:

1. 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.
2. 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 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. 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);

1. 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.

Выполнение работы:

Для выполнения первого задания я использовала следующий код:

# создание 2-х узлов:

set N 2

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

set n($i) [$ns node]

}

# соединение 2-х узлов дуплексным соединением

# с полосой пропускания 2 Мб/с и задержкой 10 мс,

# очередью с обслуживанием типа DropTail

$ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail

# создание агента UDP и присоединение его к узлу n0

set udp0 [new Agent/UDP]

$ns attach-agent $n(0) $udp0

# создание источника трафика CBR (constant bit rate)

set cbr0 [new Application/Traffic/CBR]

# устанавливаем размер пакета в 500 байт

$cbr0 set packetSize\_ 500

#задаем интервал между пакетами равным 0.005 секунды,

#т.е. 200 пакетов в секунду

$cbr0 set interval\_ 0.005

# присоединение источника трафика CBR к агенту udp0

$cbr0 attach-agent $udp0# Создание агента-приёмника и присоединение его к узлу n(1)

set null0 [new Agent/Null]

$ns attach-agent $n(1) $null0# Соединение агентов между собой

$ns connect $udp0 $null0# запуск приложения через 0,5 с

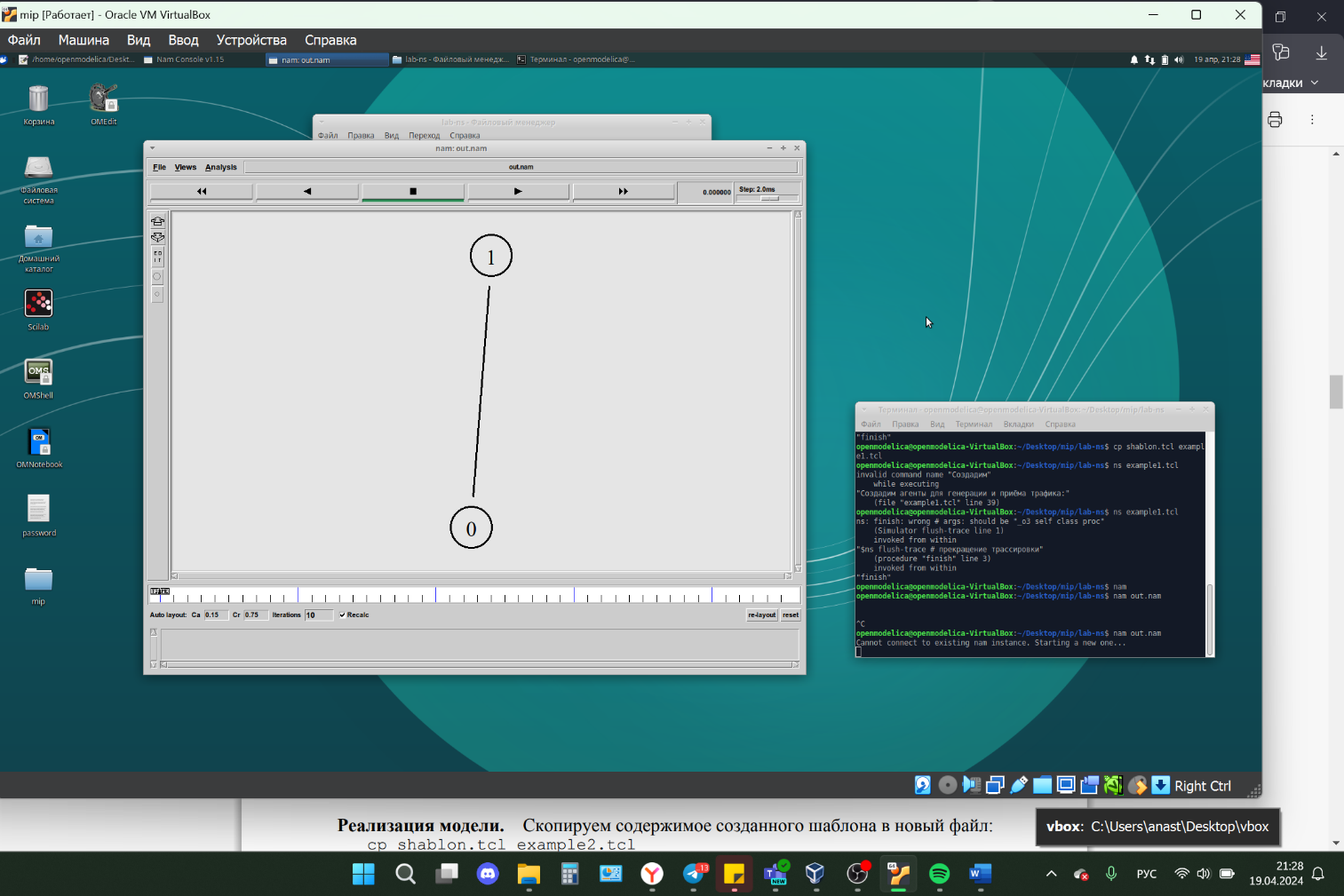
$ns at 0.5 "$cbr0 start"

# остановка приложения через 4,5 с

$ns at 4.5 "$cbr0 stop"

$ns at 5.0 "finish"

В результате я получила данный результат:



Для второго задания я использовала следующий код:

# создание объекта Simulator

set ns [new Simulator]

# открытие на запись файла out.nam для визуализатора nam

set nf [open out.nam w]

# все результаты моделирования будут записаны в переменную nf

$ns namtrace-all $nf

# открытие на запись файла трассировки out.tr

# для регистрации всех событий

set f [open out.tr w]

# все регистрируемые события будут записаны в переменную f

$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 # прекращение трассировки

close $f # закрытие файлов трассировки

close $nf # закрытие файлов трассировки nam

# запуск nam в фоновом режиме

exec nam out.nam &

exit 0

}

# создание 4-х узлов:

set N 4

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

set n($i) [$ns node]

}

$ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail

$ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail

$ns duplex-link $n(2) $n(3) 1.7Mb 20ms DropTail

$ns duplex-link-op $n(0) $n(2) orient right-down

$ns duplex-link-op $n(1) $n(2) orient right-up

$ns duplex-link-op $n(2) $n(3) orient right

## создание агента UDP и присоединение его к узлу n(0)

set udp0 [new Agent/UDP]

$ns attach-agent $n(0) $udp0

# создание источника CBR-трафика

# и присоединение его к агенту udp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.005

$cbr0 attach-agent $udp0

# создание агента TCP и присоединение его к узлу n(1)

set tcp1 [new Agent/TCP]

$ns attach-agent $n(1) $tcp1

# создание приложения FTP

# и присоединение его к агенту tcp1

set ftp [new Application/FTP]

$ftp attach-agent $tcp1

# создание агента-получателя для udp0

set null0 [new Agent/Null]

$ns attach-agent $n(3) $null0

# создание агента-получателя для tcp1

set sink1 [new Agent/TCPSink]

$ns attach-agent $n(3) $sink1

$ns connect $udp0 $null0

$ns connect $tcp1 $sink1

$ns color 1 Blue

$ns color 2 Red

$udp0 set class\_ 1

$tcp1 set class\_ 2

$ns duplex-link-op $n(2) $n(3) queuePos 0.5

$ns queue-limit $n(2) $n(3) 20

$ns at 0.5 "$cbr0 start"

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 4.5 "$cbr0 stop"

# at-событие для планировщика событий, которое запускает

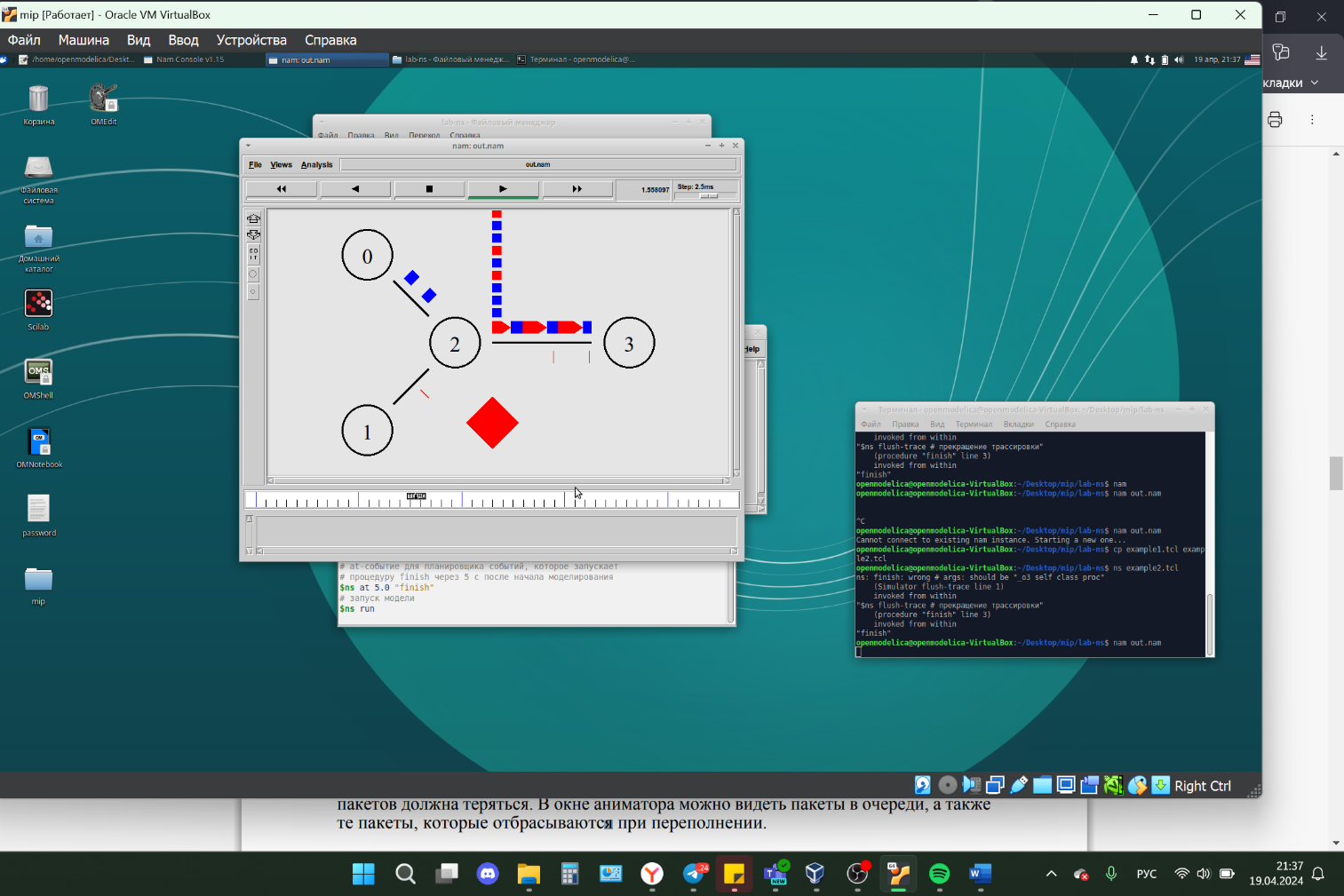
# процедуру finish через 5 с после начала моделирования

$ns at 5.0 "finish"

# запуск модели

$ns run

В результате я получила следующую модель:



Для третьего задания я использовала следующий код:

set N 7

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

set n($i) [$ns node]

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

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

}set udp0 [new Agent/UDP]

$ns attach-agent $n(0) $udp0

set cbr0 [new Agent/CBR]

$ns attach-agent $n(0) $cbr0

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.005

set null0 [new Agent/Null]

$ns attach-agent $n(3) $null0

$ns connect $cbr0 $null0

$ns at 0.5 "$cbr0 start"

$ns rtmodel-at 1.0 down $n(1) $n(2)

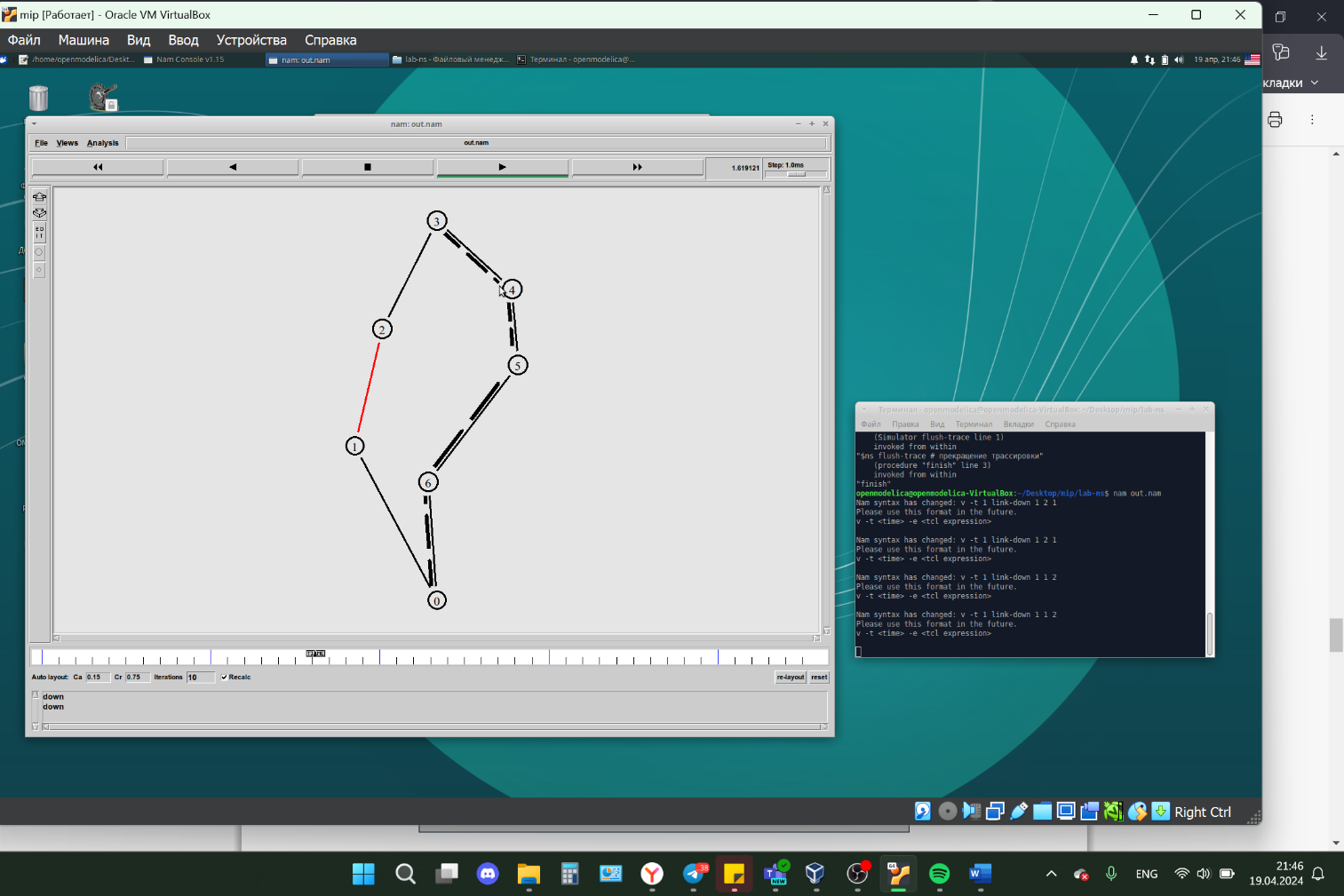
$ns rtmodel-at 2.0 up $n(1) $n(2)

$ns at 4.5 "$cbr0 stop"

$ns rtproto DV

$ns at 5.0 "finish"

В результате я получила следующую картину (скриншот был сделан в момент разрыва между узлами n(1) и n(2):



Для последнего задания я написала код:

# создание объекта Simulator

set ns [new Simulator]

# открытие на запись файла out.nam для визуализатора nam

set nf [open out.nam w]

# все результаты моделирования будут записаны в переменную nf

$ns namtrace-all $nf

# открытие на запись файла трассировки out.tr

# для регистрации всех событий

set f [open out.tr w]

# все регистрируемые события будут записаны в переменную f

$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 # прекращение трассировки

close $f # закрытие файлов трассировки

close $nf # закрытие файлов трассировки nam

# запуск nam в фоновом режиме

exec nam out.nam &

exit 0

}

set N 6

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

set n($i) [$ns node]

}

$ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail

$ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail

$ns duplex-link $n(1) $n(5) 2Mb 10ms DropTail

$ns duplex-link $n(2) $n(3) 2Mb 10ms DropTail

$ns duplex-link $n(3) $n(4) 2Mb 10ms DropTail

$ns duplex-link $n(4) $n(0) 2Mb 10ms DropTail

set tcp [new Agent/TCP/Newreno]

$ns attach-agent $n(0) $tcp

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set packetSize 500

$ftp set Interval\_ 0.005

set sink [new Agent/TCPSink/DelAck]

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

$ns connect $tcp $sink

$ns at 0.5 "$ftp start"

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

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

$ns at 4.5 "$ftp stop"

$ns rtproto DV

# at-событие для планировщика событий, которое запускает

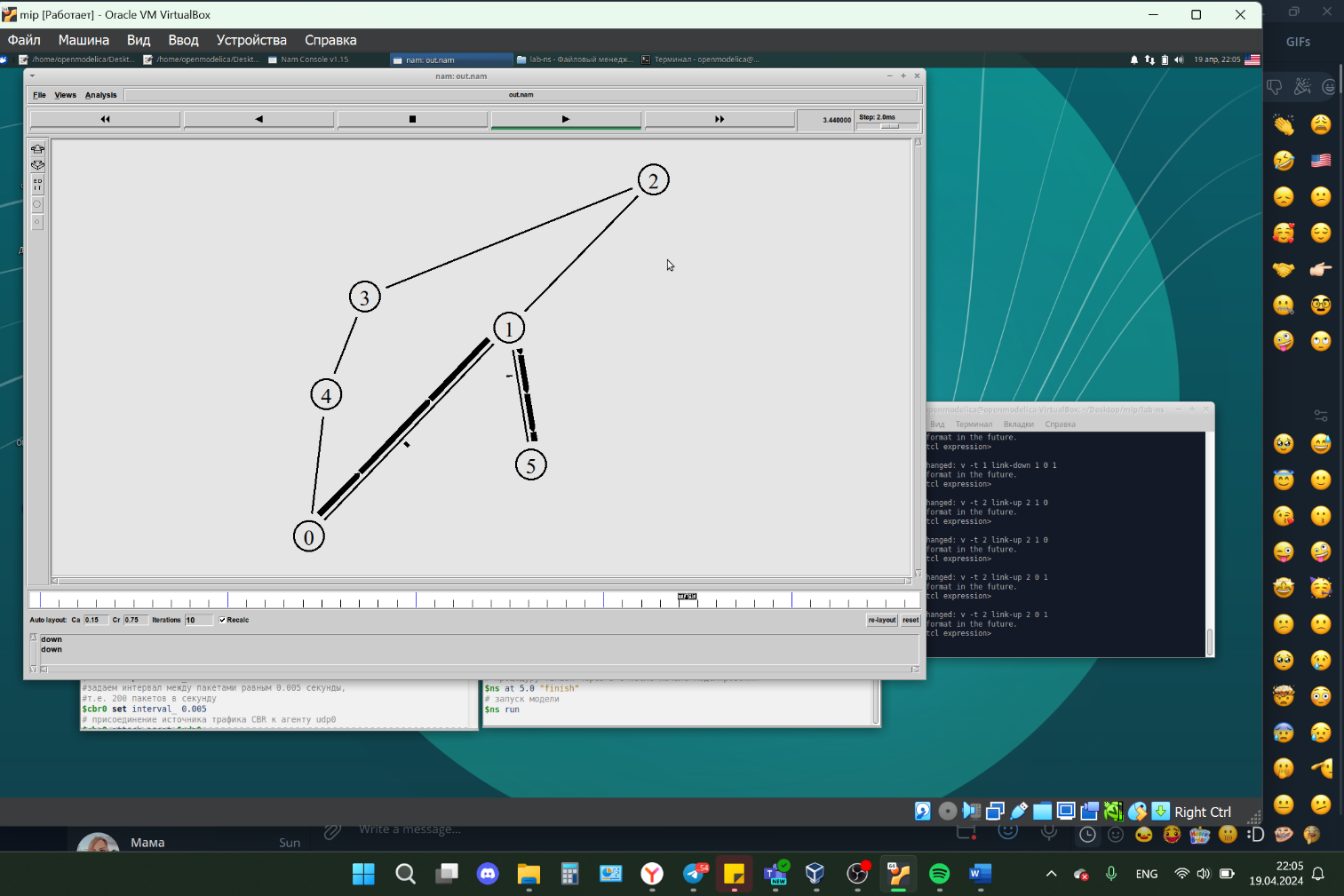
# процедуру finish через 5 с после начала моделирования

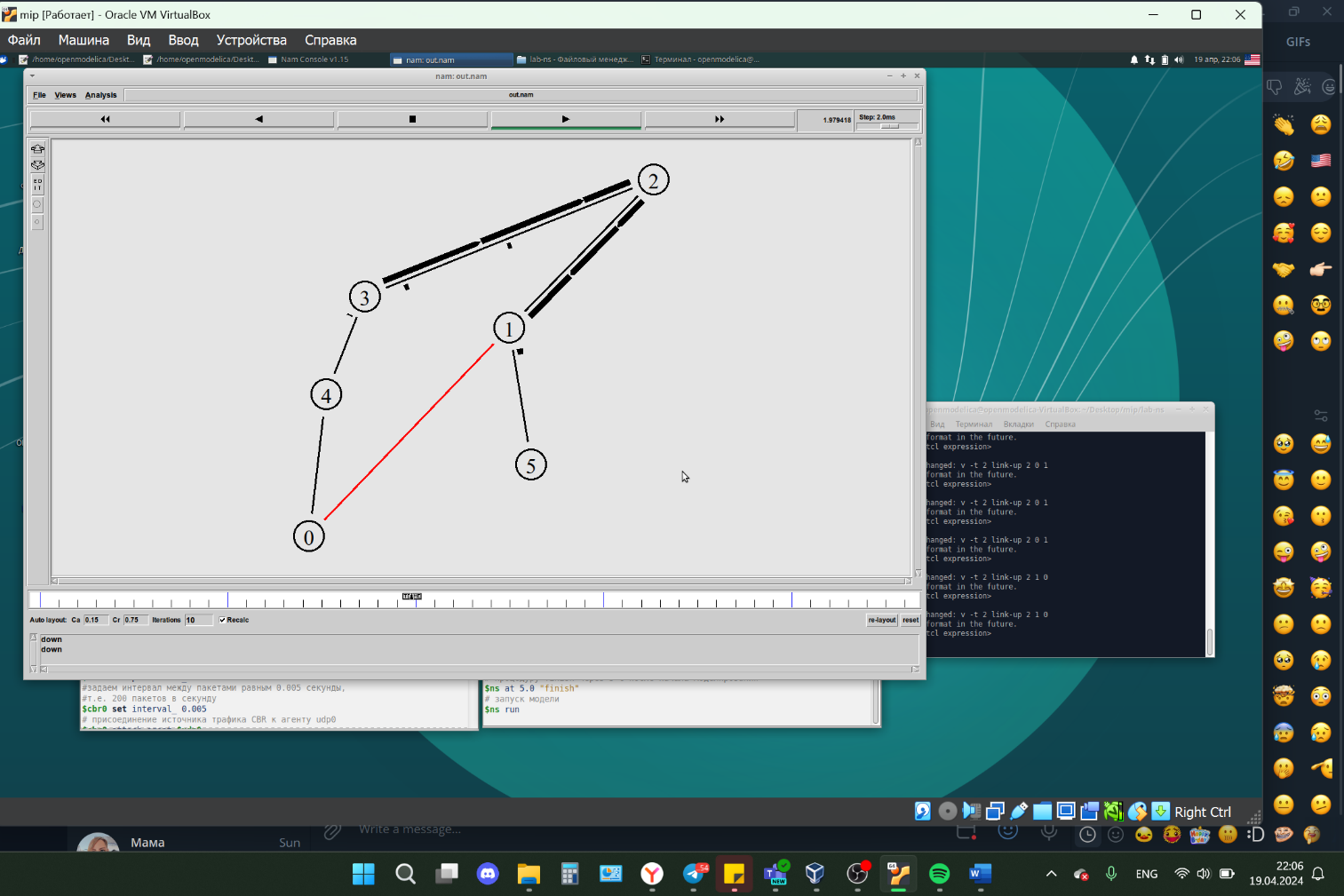
$ns at 5.0 "finish"

# запуск модели

$ns run

В результате получила данную модель:





Вывод: в ходе работы я приобрела навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2.