

# Лабораторная работа №1

Простые модели компьютерной сети

---

Дворкина Е. В.

14 февраля 2025

Российский университет дружбы народов, Москва, Россия

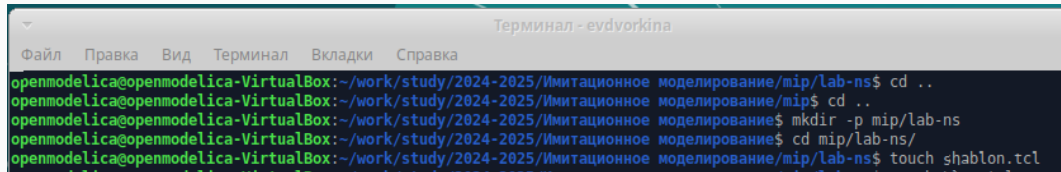
- Дворкина Ева Владимировна
- студентка
- группа НФИбд-01-22
- Российский университет дружбы народов
- 1132226447@rudn.ru
- <https://github.com/evdvorkina>



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

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 усложненной топологией сети
4. Выполнить пример с кольцевой топологией сети
5. Выполнить упражнение

Создание директории `mip/lab-ns` и файла `shablon.tcl`.



```
Терминал - evdvorkina
Файл  Правка  Вид  Терминал  Вкладки  Справка
openmodelica@openmodelica-VirtualBox:~/work/study/2024-2025/Имитационное моделирование/mip/lab-ns$ cd ..
openmodelica@openmodelica-VirtualBox:~/work/study/2024-2025/Имитационное моделирование/mip$ cd ..
openmodelica@openmodelica-VirtualBox:~/work/study/2024-2025/Имитационное моделирование$ mkdir -p mip/lab-ns
openmodelica@openmodelica-VirtualBox:~/work/study/2024-2025/Имитационное моделирование$ cd mip/lab-ns/
openmodelica@openmodelica-VirtualBox:~/work/study/2024-2025/Имитационное моделирование/mip/lab-ns$ touch shablon.tcl
```

Рис. 1: Создание директории и файла шаблона

## Шаблон сценария для NS-2

```
▼ /home/openmodelica/work/study/2024-2025/Имитационное моделирование/mip/lab-ns — + ×
Файл  Правка  Поиск  Вид  Документ  Справка

# создание объекта Simulator
set ns [new Simulator]

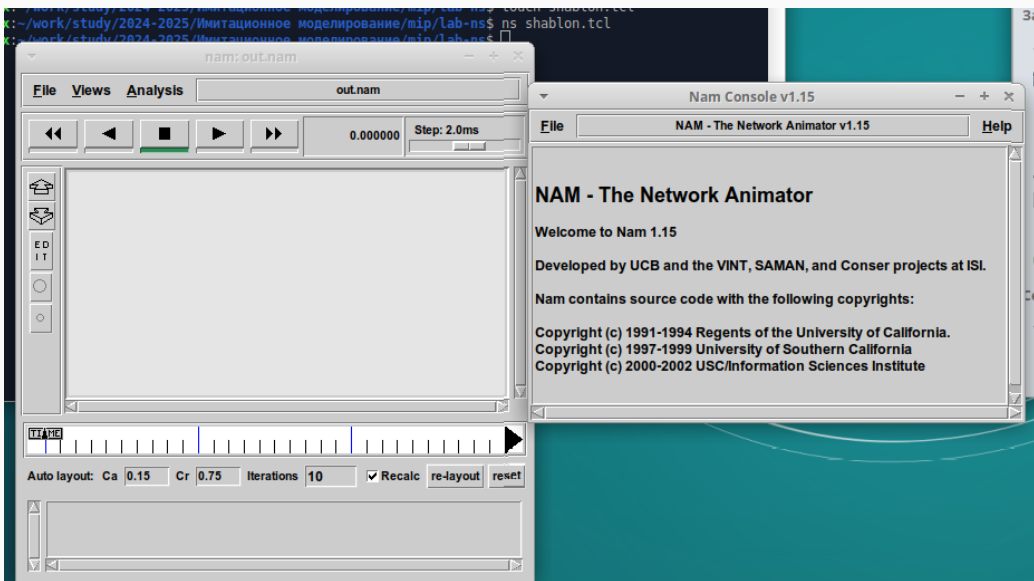
# открытие на запись файла out.nam для визуализатора nam
set nf [open out.nam w]
# все результаты моделирования будут записаны в переменную nf
$ns namtrace-all $nf

# 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
    # прекращение трассировки
    $ns flush-trace
    # закрытие файлов трассировки
    close $f
    # закрытие файлов трассировки nam
    close $nf
    # 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 модели
$ns run
```

## Шаблон сценария для NS-2



**Постановка задачи.** 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.



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

```

* /home/openmodelica/work/study/2024-2025/Имитационное моделирование/mip/lab-r - + x
Файл Правка Поиск Вид Документ Справка
}

# 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

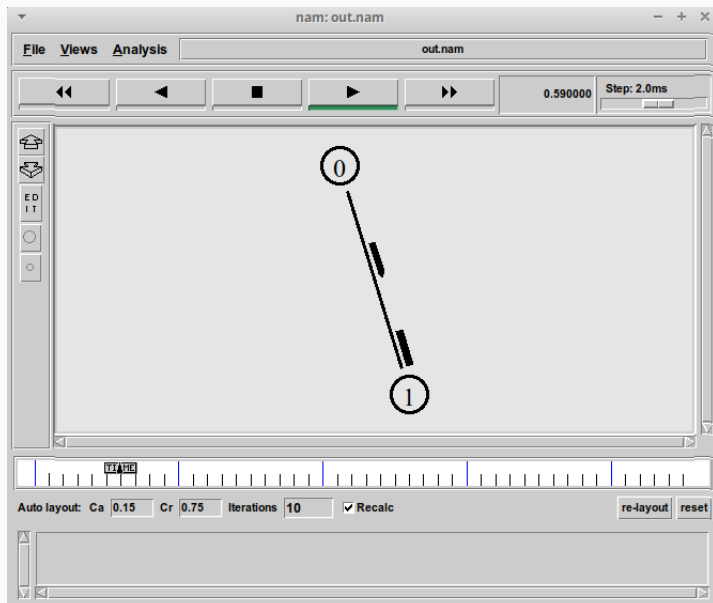
# создание агента UDP и присоединение его к узлу n0
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
# создание источника трафика CBR (constant bit rate)
set cbr0 [new Application/Traffic/CBR]
# устанавливаем размер пакета в 500 байт
$cbr0 set packetSize_ 500
# задаем интервал между пакетами равным 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

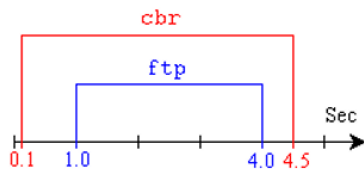
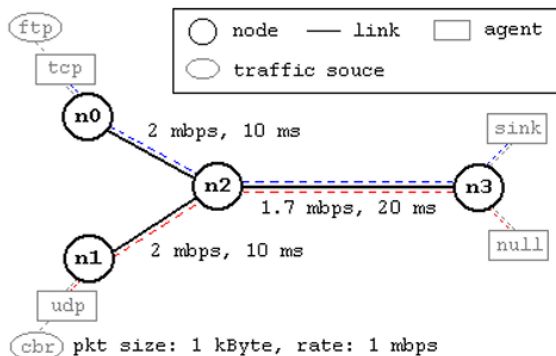
# Соединение агентов между собой
$ns connect $udp0 $null0

# запуск приложения через 0,5 с
$ns at 0.5 "$cbr0 start"
# остановка приложения через 4,5 с
$ns at 4.5 "$cbr0 stop"
```

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



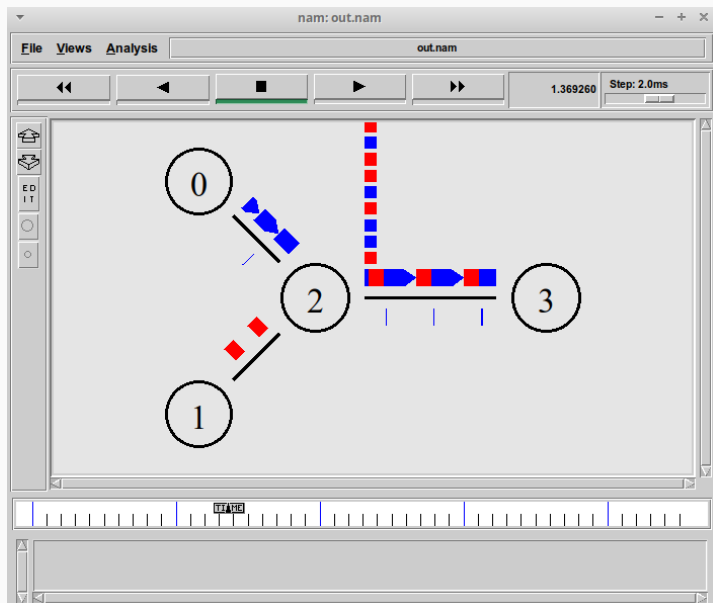
## Пример с усложненной топологией сети



## Пример с усложненной топологией сети

```
31 set N 4
32 for {set i 0} {$i < $N} {incr i} {
33     set n($i) [$ns node]
34 }
35 $ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail
36 $ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail
37 $ns duplex-link $n(3) $n(2) 1.7Mb 20ms DropTail
38 $ns duplex-link-op $n(0) $n(2) orient right-down
39 $ns duplex-link-op $n(1) $n(2) orient right-up
40 $ns duplex-link-op $n(2) $n(3) orient right
41
42 # создание агента UDP и присоединение его к узлу n(0)
43 set udp0 [new Agent/UDP]
44 $ns attach-agent $n(1) $udp0
45 # создание источника CBR-трафика
46 # и присоединение его к агенту udp0
47 set cbr0 [new Application/Traffic/CBR]
48 $cbr0 set packetSize 500
49 $cbr0 set interval 0.005
50 $cbr0 attach-agent $udp0
51
52 # создание агента TCP и присоединение его к узлу n(1)
53 set tcp1 [new Agent/TCP]
54 $ns attach-agent $n(0) $tcp1
55 # создание приложения FTP
56 # и присоединение его к агенту tcp1
57 set ftp [new Application/FTP]
58 $ftp attach-agent $tcp1
59
60 # создание агента-получателя для udp0
61 set null0 [new Agent/Null]
62 $ns attach-agent $n(3) $null0
63 # создание агента-получателя для tcp1
64 set sink1 [new Agent/TCPSink]
65 $ns attach-agent $n(3) $sink1
66
67 $ns connect $udp0 $null0
68 $ns connect $tcp1 $sink1
69
70 $ns color 1 Blue
71 $ns color 2 Red
72 $udp0 set class 2
73 $tcp1 set class 1
74
75 $ns duplex-link-op $n(2) $n(3) queuePos 0.5
76 $ns queue-limit $n(2) $n(3) 20
77
78 $ns at 0.1 "$cbr0 start"
79 $ns at 1.0 "$ftp start"
80 $ns at 4.0 "$ftp stop"
81 $ns at 4.5 "$cbr0 stop"
```

## Пример с усложненной топологией сети



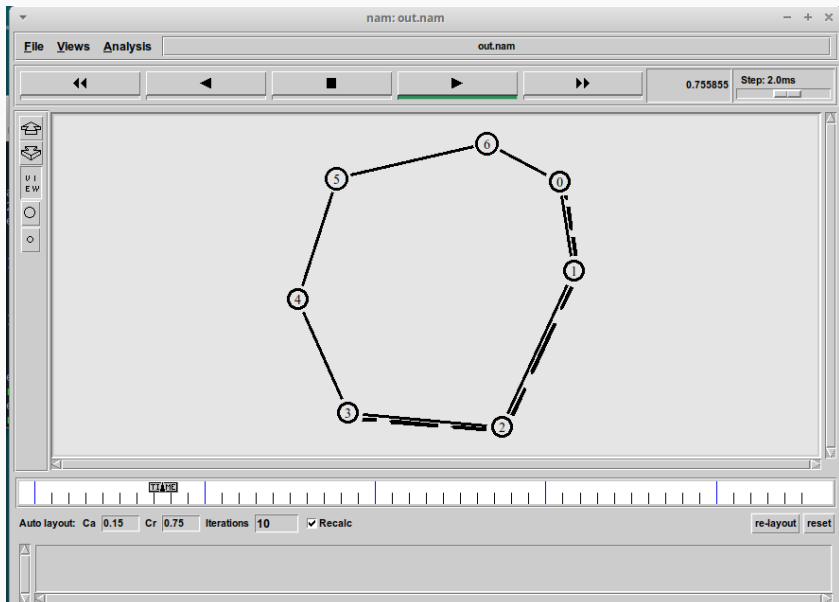
### Постановка задачи.

- 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

## Пример с кольцевой топологией сети

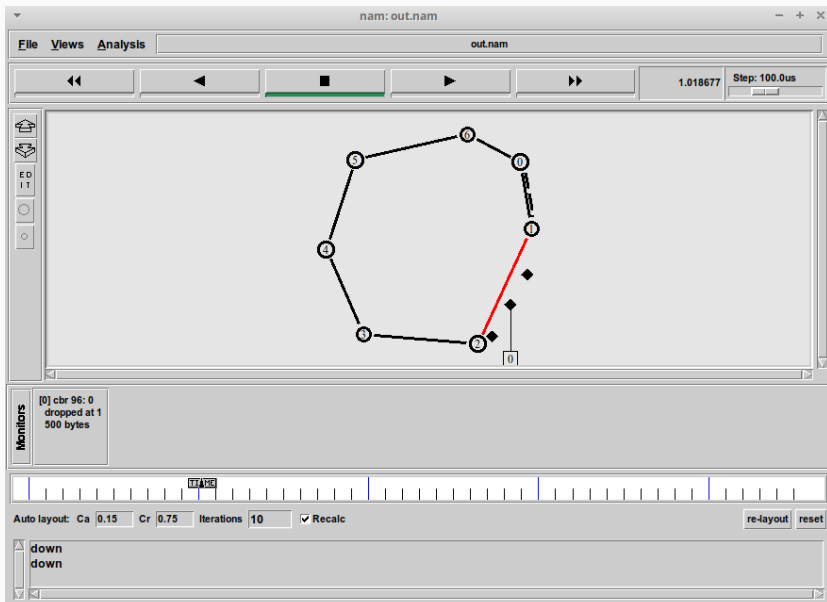
```
31 # 7 узлов
32 set N 7
33 for {set i 0} {$i < $N} {incr i} {
34   set n($i) [$ns node]
35 }
36 # Далее соединим узлы так, чтобы создать круговую топологию:
37 for {set i 0} {$i < $N} {incr i} {
38   $ns duplex-link $n($i) $n([expr ($i+1)%$N]) 1Mb 10ms DropTail
39 }
40 #Каждыйузел, за исключением последнего, соединяется со следующим, последний
41 #соединяется с первым. Для этого в цикле использован оператор %, означающий
42 #остаток от деления нацело.
43 #Зададим передачу данных от узла n(0) к узлу n(3):
44 set udp0 [new Agent/UDP]
45 $ns attach-agent $n(0) $udp0
46 set cbr0 [new Agent/CBR]
47 $ns attach-agent $n(0) $cbr0
48 $cbr0 set packetSize_ 500
49 $cbr0 set interval_ 0.005
50 set null0 [new Agent/Null]
51 $ns attach-agent $n(3) $null0
52 $ns connect $cbr0 $null0
53
54 $ns at 0.5 "$cbr0 start"
55 #разрыв между узлами 1 и 2 на 1 секунду
56 $ns rtmodel-at 1.0 down $n(1) $n(2)
57 $ns rtmodel-at 2.0 up $n(1) $n(2)
58 $ns at 4.5 "$cbr0 stop"
```

## Пример с кольцевой топологией сети





## Пример с кольцевой топологией сети



## Пример с кольцевой топологией сети

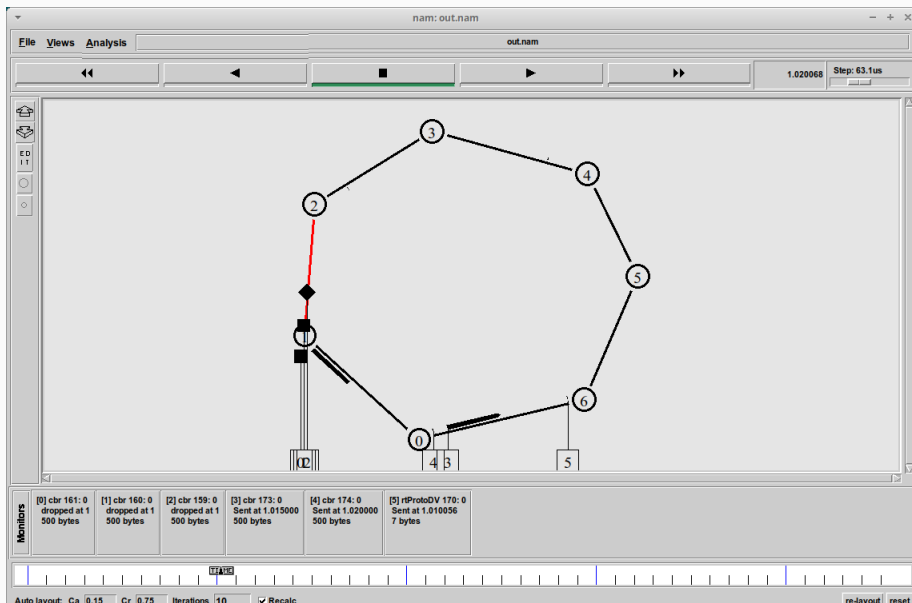
Добавим в начало скрипта после команды создания объекта Simulator: `$ns rtproto DV`

Сразу после запуска в сети отправляется небольшое количество маленьких пакетов, используемых для обмена информацией, необходимой для маршрутизации между узлами. Когда соединение будет разорвано, информация о топологии будет обновлена

```
1 # создание объекта Simulator
2 set ns [new Simulator]
3
4 $ns rtproto DV
5 |
```

Рис. 12: Описание кольцевой топологии сети с динамической маршрутизацией пакетов

# Пример с кольцевой топологией сети

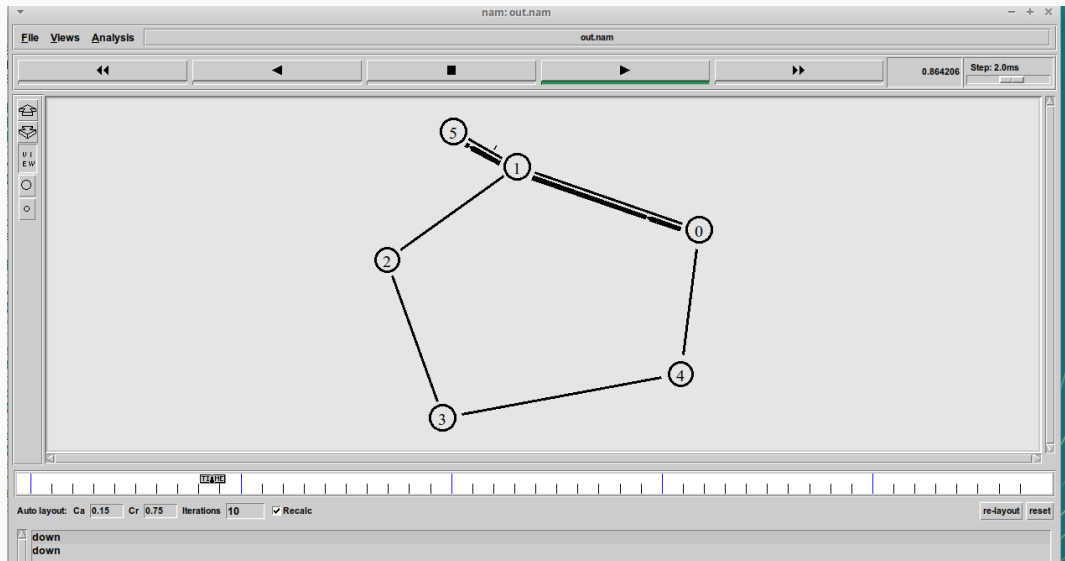


1. Изменим количество узлов в кольце на 5, узел  $n_5$  оподсоединим к узлу  $n(1)$
2. Создадим агента TCP типа Newreno, а на принимающей стороне зададим TCPSink-объект типа DelAck. Поверх TCP работает протокол FTP
3. Протокол FTP работает с 0,5 до 4,5 секунд модельного времени
4. Зададим разрыв соединения между узлами  $n(0)$  и  $n(1)$  с 1 по 2 секунду модельного времени.

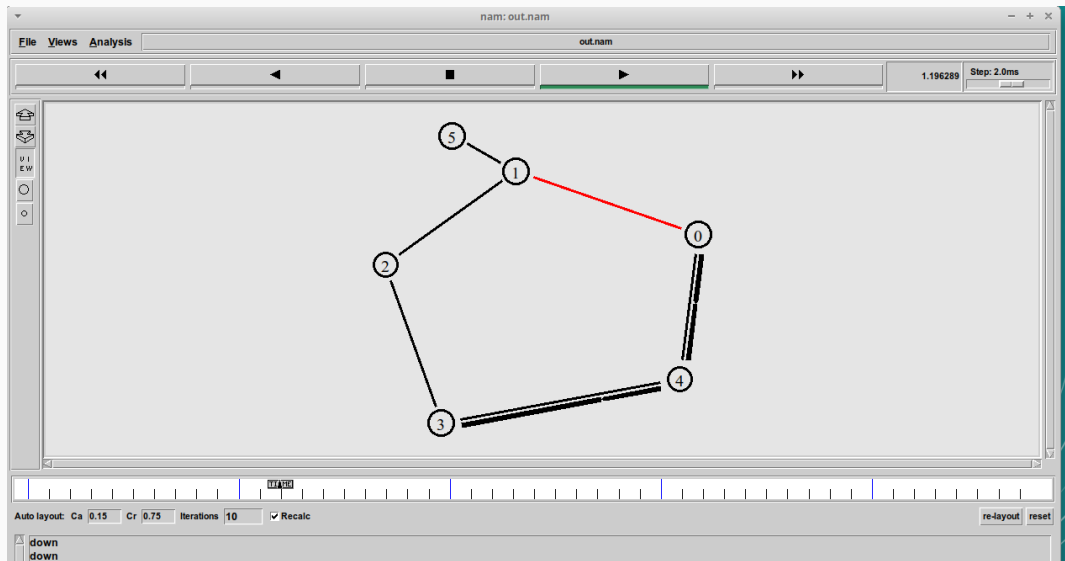
## Упражнение

```
32
33 set N 5
34 for {set i 0} {$i < $N} {incr i} {
35     set n($i) [$ns node]
36 }
37
38 for {set i 0} {$i < $N} {incr i} {
39     $ns duplex-link $n($i) $n([expr ($i+1)%$N]) 1Mb 10ms DropTail
40 }
41
42 set n5 [$ns node]
43 $ns duplex-link $n5 $n(1) 1Mb 10ms DropTail
44
45 set tcp1 [new Agent/TCP/Newreno]
46 $ns attach-agent $n(0) $tcp1
47
48 set ftp [new Application/FTP]
49 $ftp attach-agent $tcp1
50
51 set sink1 [new Agent/TCPSink/DelAck]
52 $ns attach-agent $n5 $sink1
53 $ns connect $tcp1 $sink1
54
55 $ns at 0.5 "$ftp start"
56 $ns rtmodel-at 1.0 down $n(0) $n(1)
57 $ns rtmodel-at 2.0 up $n(0) $n(1)
58 $ns at 4.5 "$ftp stop"
59
```

# Упражнение



# Упражнение



При выполнении данной лабораторной работы я приобрела навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также провела анализ полученных результатов моделирования.



Спасибо за внимание!

---