

# Лабораторная работа 1

Простые модели компьютерной сети

---

Туем Г.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Туем Гислен
- Студент
- НКНбд-01-22
- Российский университет дружбы народов
- 1032225069@pfur.ru



## Цели и задачи

---

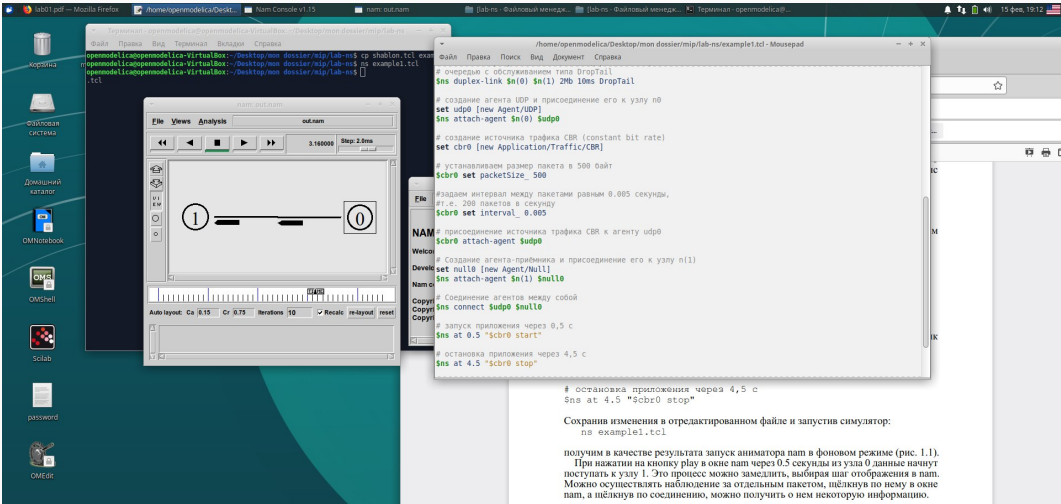
Приобретить навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов моделирования.

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 усложнённой топологией сети;
4. Выполнить пример с кольцевой топологией сети;
5. Выполнить упражнение.

## Выполнение лабораторной работы

---

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



The screenshot shows a virtual machine environment with a desktop background. Several windows are open:

- Terminal:** Displays the configuration of a network topology. The configuration includes:
  - Creation of a UDP agent and its attachment to node n0.
  - Creation of a CBR (constant bit rate) traffic source and its attachment to node n0.
  - Setting of packet size to 500 bytes and interval to 0.005 seconds.
  - Creation of a null agent and its attachment to node n1.
  - Connection of the agents between nodes n0 and n1.
  - Execution of the application at 0.5 seconds and stopping it at 4.5 seconds.
- NAM (Network Animator):** Shows a visual representation of the network topology. It displays two nodes, labeled 1 and 0, connected by a line. The interface includes a timeline and various controls.

Сохранив изменения в отредактированном файле и запустив симулятор:  
ns example1.tcl

получим в качестве результата запуск аниматора nam в фоновом режиме (рис. 1.1).  
При нажатии на кнопку play в окне nam через 0.5 секунды из узла 0 данные начнут поступать к узлу 1. Это процесс можно замедлить, выбирая шаг отображения в nam. Можно осуществлять наблюдение за отдельным пакетом, щёлкнув по нему в окне nam, а щёлкнув по соединению, можно получить о нем некоторую информацию.

## Выполнение лабораторной работы

---



# Выполнение лабораторной работы

## Пример с усложнённой топологией сети

The screenshot displays a computer screen with multiple windows. The primary window is a terminal running a NetMiner script. The script configures a network topology with four nodes (0, 1, 2, 3) and various links. It also sets up agents for UDP, TCP, and FTP traffic. The terminal output shows the script execution progress, including the creation of agents and the start of the simulation. Overlaid on the terminal are two windows: a Firefox browser showing a PDF document and a NetMiner GUI window displaying a network diagram with four nodes and their connections. The GUI window also shows a timeline and a status bar.

```
$ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(3) $n(2) 2Mb 10ms 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
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
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
# создание приложения FTP
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"
```

Firefox browser window shows a PDF document with the URL [https://esystem.rudn.ru/pluginfile.php/2585540/mod\\_resource/content/4/lab01.pdf?forcedownload=1](https://esystem.rudn.ru/pluginfile.php/2585540/mod_resource/content/4/lab01.pdf?forcedownload=1).

NetMiner GUI window displays a network diagram with four nodes (0, 1, 2, 3) and their connections. The diagram shows a central node 2 connected to nodes 0, 1, and 3. Node 0 is connected to node 2, node 1 is connected to node 2, and node 3 is connected to node 2. The GUI window also shows a timeline and a status bar.

## Выполнение лабораторной работы

---

# Выполнение лабораторной работы

## Пример с кольцевой топологией сети

The screenshot shows a desktop environment with several windows. The primary window is a NetLogo simulation titled 'out.nam'. It displays a ring network topology with 8 nodes connected in a loop. The nodes are represented by small circles, and the connections are lines. A status bar at the bottom of the NetLogo window shows 'Auto layout: Ca 0.15 Cr 0.75 Iterations 10' and 'Recalc re-layout reset' buttons.

On the left, a terminal window shows the following commands:

```
# создание объекта Simulator
set ns [new Simulator]
$ns rtproto DV

# открытие на запись файла out.nam для визуализации
set nf [open out.nam w]
# все результаты моделирования будут записаны в файл
$ns namtrace-all $nf
# открытие на запись файла трассировки out.tr
# для регистрации всех событий
set f [open out.tr w]

# процедура finish закрывает файлы трассировки
# и запускает визуализатор nam
proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

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])
}

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 at 5.0 "finish"
```

Below the terminal window, a smaller window titled 'NAM - The Network' is visible. It contains the text: 'Welcome to NAM 1.15', 'Developed by UCB and the V...', 'Nam contains source code w...', 'Copyright (c) 1991-1994 Raga...', 'Copyright (c) 1997-1999 Univ...', 'Copyright (c) 2000-2002 USC...', and a diagram of a ring network topology.

On the right, a web browser window shows a URL: [https://esystem.rudn.ru/pluginfile.php/2585540/mod\\_resource/content/4/lab01.pdf?forcedownload=1](https://esystem.rudn.ru/pluginfile.php/2585540/mod_resource/content/4/lab01.pdf?forcedownload=1). The page content is partially visible, showing text about network simulation and a diagram of a ring network topology.

## Выполнение лабораторной работы

---

# Выполнение лабораторной работы

Когда соединение будет разорвано, информация о топологии будет обновлена, и пакеты будут отправляться по новому маршруту через узлы n(6), n(5) и n(4)

The screenshot displays the NAM (Network Animator) environment. The main window shows a network topology diagram with nodes n(1) through n(7) and links. A terminal window in the foreground shows the command-line interface for the simulator, including commands like 'set ns', 'set nf', 'set f', and 'proc finish'. A smaller window titled 'NAM - The Network Animator' shows the same topology diagram. The background shows a web browser window with a URL from esystem.rudn.ru.

```
# создание объекта Simulator
set ns [new Simulator]
$ns rtproto DV

# открытие файла для записи out.nam для визуализации
set nf [open out.nam w]
# все результаты моделирования будут записаны в файл
$ns nantrace-all $nf
# открытие файла трассировки out.tr
# для регистрации всех событий
set f [open out.tr w]

# процедура finish закрывает файлы трассировки
# и запускает визуализатор nam
proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

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}]) 1000000 1000000 0
}

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 at 5.0 "finish"
```

Скриншоты интерфейсов:

- Terminal:** Показывает команды и сообщения о состоянии симуляции.
- NAM - The Network Animator:** Отображает топологию сети с узлами n(1) и n(2) и их соединением.
- Web Browser:** Показывает страницу с информацией о системе.

## Упражнение

---

# Упражнение

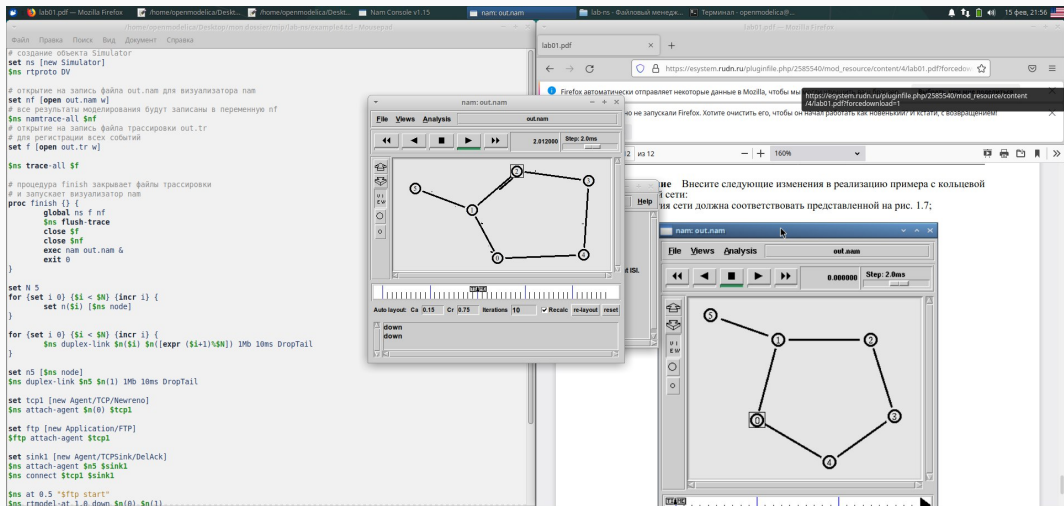


Рис. 5: Изменённая кольцевая топология сети

## Упражнение

---



# Упражнение

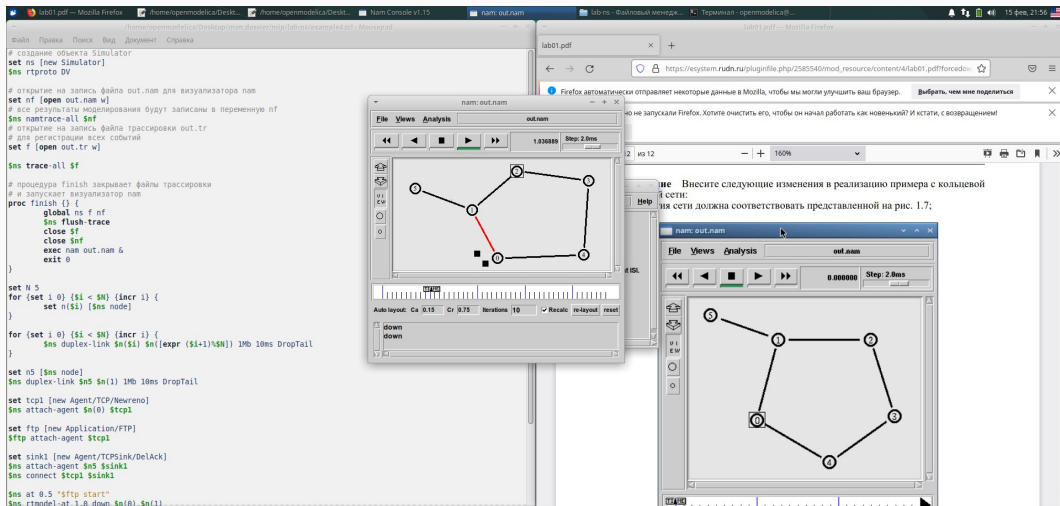


Рис. 6: Разрыв

## 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.