

## Лабораторная работа № 8

---

Ли Тимофей Александрович, НФИбд-01-18

## Цель работы

---

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

## Задачи

---

- Сделать примеры заданий по NS-2
- Выполнить задание

## Ход работы

---

## 1.1 Шаблон сценария для NS-2

Установил виртуальную машину с ОС Ubuntu, на нее установил Nam и Ns2. Затем, в рабочем каталоге создал папку `mir` для лабораторных работ, а в ней папку `lab-ns`. Создал файл `shablon.tcl` и написал в нем следующий код (рис. @fig:001):

```
1 set ns [new Simulator]
2 set nf [open out.nam w]
3 $ns namtrace-all $nf
4 set f [open out.tr w]
5 $ns trace-all $f
6 proc finish {} {
7     global ns f nf
8     $ns flush-trace
9     close $f
10    close $nf
11    exec nam out.nam &
12    exit 0
13 }
14 $ns at 5.0 "finish"
15 $ns run
```

Рис. 1: shablon.tcl

## 1.2 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.

Для решения задачи скопировал содержимое `shablon.tcl` в `example1.tcl` и написал следующий код: (рис. @fig:002)

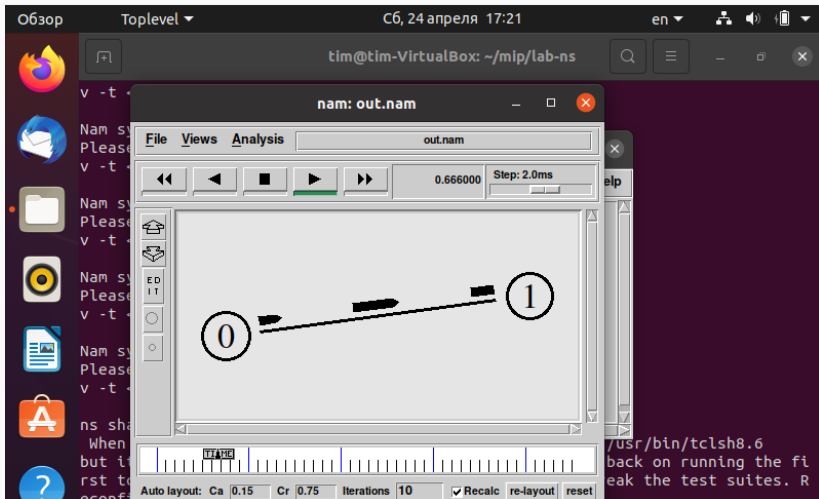


## 1.2 Простой пример описания топологии сети, состоящей из двух узлов и одного соединения

```
1 set ns [new Simulator]
2 set nf [open out.nam w]
3 $ns namtrace-all $nf
4 set f [open out.tr w]
5 $ns trace-all $f
6 proc finish {} {
7     global ns f nf
8     $ns flush-trace
9     close $f
10    close $nf
11    exec nam out.nam &
12    exit 0
13 }
14 set N 2
15 for {set i 0} {$i < $N} {incr i} {
16     set n($i) [$ns node]
17 }
18 $ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail
19 set udp0 [new Agent/UDP]
20 $ns attach-agent $n(0) $udp0
21 set cbr0 [new Application/Traffic/CBR]
22 $cbr0 set packetSize_ 500
23 $cbr0 set interval_ 0.005
24 $cbr0 attach-agent $udp0
25 set null0 [new Agent/Null]
26 $ns attach-agent $n(1) $null0
27 $ns connect $udp0 $null0
28 $ns at 0.5 "$cbr0 start"
29 $ns at 4.5 "$cbr0 stop"
```

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

Получилась следующая модель: (рис. @fig:003)



## 1.3 Пример с усложненной топологией сети

Задача: требуется построить модель по следующим критериям:

- сеть состоит из 4 узлов ( $n_0$ ,  $n_1$ ,  $n_2$ ,  $n_3$ );
- между узлами  $n_0$  и  $n_2$ ,  $n_1$  и  $n_2$  установлено дуплексное соединение с пропускной способностью 2 Мбит/с и задержкой 10 мс;
- между узлами  $n_2$  и  $n_3$  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  $n_0$  подключается к TCP-приёмнику на узле  $n_3$  (по-умолчанию, максимальный размер пакета, который TCP-агент может генерировать, равняется 1KByte)
- TCP-приёмник генерирует и отправляет ACK пакеты отправителю и откидывает полученные пакеты;

## 1.3 Пример с усложненной топологией сети

- 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.

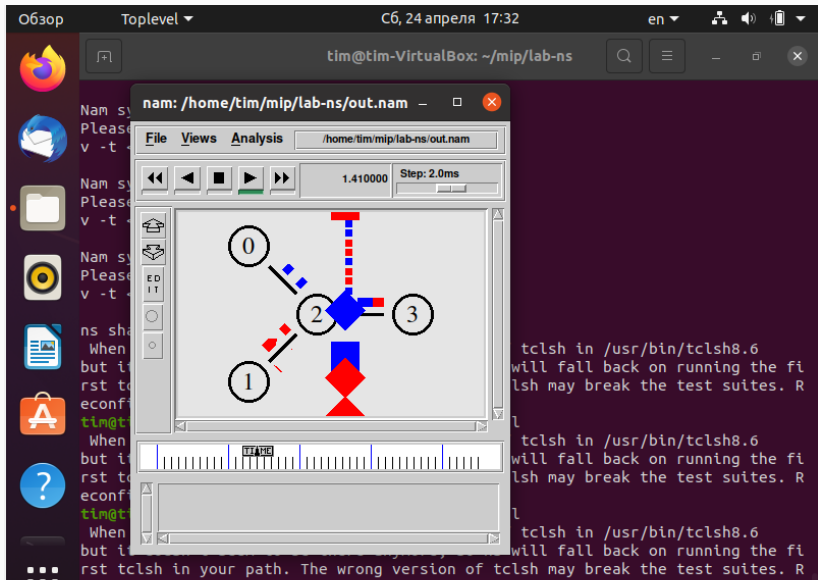
Для решения задачи скопировал содержимое shablon.tcl в example2.tcl и написал следующий код: (рис. @fig:004)

## 1.3 Пример с усложненной топологией сети

```
1 set ns [new Simulator]
2 set nf [open out.nam w]
3 $ns namtrace-all $nf
4 set f [open out.tr w]
5 $ns trace-all $f
6 proc finish {} {
7     global ns f nf
8     $ns flush-trace
9     close $f
10    close $nf
11    exec nam out.nam &
12    exit 0
13 }
14 set N 4
15 for {set i 0} {$i < $N} {incr i} {
16     set n($i) [$ns node]
17 }
18 $ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail
19 $ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail
20 $ns duplex-link $n(3) $n(2) 2Mb 10ms DropTail
21 $ns duplex-link-op $n(0) $n(2) orient right-down
22 $ns duplex-link-op $n(1) $n(2) orient right-up
23 $ns duplex-link-op $n(2) $n(3) orient right
24 set udp0 [new Agent/UDP]
25 $ns attach-agent $n(0) $udp0
26 set cbr0 [new Application/TrafficCBR]
27 $cbr0 set packetSize_ 500
28 $cbr0 set interval_ 0.005
29 $cbr0 attach-agent $udp0
30 set tcp1 [new Agent/TCP]
31 $ns attach-agent $n(1) $tcp1
32 set ftp [new Application/FTP]
33 $ftp attach-agent $tcp1
34 set null0 [new Agent/Null]
35 $ns attach-agent $n(3) $null0
36 set sink1 [new Agent/TCPSink]
37 $ns attach-agent $n(3) $sink1
38 $ns connect $udp0 $null0
39 $ns connect $tcp1 $sink1
40 $ns color 1 Blue
41 $ns color 2 Red
42 $udp0 set class_ 1
43 $tcp1 set class_ 2
44 $ns duplex-link-op $n(2) $n(3) queuePos 0.5
45 $ns queue-limit $n(2) $n(3) 20
46 $ns at 0.5 "$cbr0 start"
47 $ns at 1.0 "$ftp start"
48 $ns at 4.0 "$ftp stop"
49 $ns at 4.5 "$cbr0 stop"
50 $ns at 5.0 "finish"
```

### 1.3 Пример с усложненной топологией сети

Результат: (рис. @fig:005)



## 1.4 Пример с кольцевой топологией сети

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.

Для решения задачи скопировал содержимое `shablon.tcl` в `example3.tcl` и написал следующий код: (рис. @fig:006)

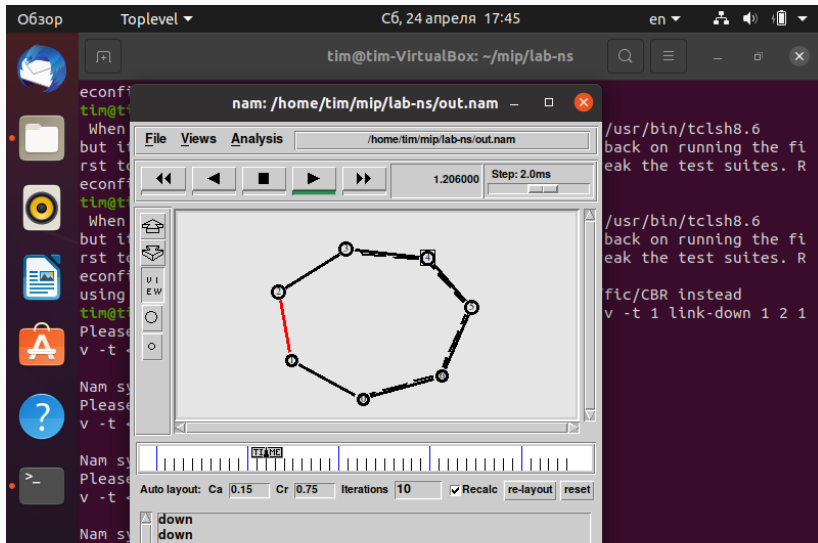
## 1.4 Пример с кольцевой топологией сети

```
1 set ns [new Simulator]
2 $ns rtproto DV
3 set nf [open out.nam w]
4 $ns namtrace-all $nf
5 set f [open out.tr w]
6 $ns trace-all $f
7 proc finish {} {
8     global ns f nf
9     $ns flush-trace
10    close $f
11    close $nf
12    exec nam out.nam &
13    exit 0
14 }
15
16 set N 7
17 for {set i 0} {$i < $N} {incr i} {
18     set n($i) [$ns node]
19 }
20 for {set i 0} {$i < $N} {incr i} {
21     $ns duplex-link $n($i) $n([expr ($i+1)%$N]) 1Mb 10ms DropTail
22 }
23 set udp0 [new Agent/UDP]
24 $ns attach-agent $n(0) $udp0
25 set cbr0 [new Agent/CBR]
26 $ns attach-agent $n(0) $cbr0
27 $cbr0 set packetSize_ 500
28 $cbr0 set interval_ 0.005
29 set null0 [new Agent/Null]
30 $ns attach-agent $n(3) $null0
31 $ns connect $cbr0 $null0
32 $ns at 0.5 "$cbr0 start"
33 $ns rtmodel-at 1.0 down $n(1) $n(2)
34 $ns rtmodel-at 2.0 up $n(1) $n(2)
35 $ns at 4.5 "$cbr0 stop"
36 $ns at 5.0 "finish"
```



## 1.4 Пример с кольцевой топологией сети

Результат: (рис. @fig:007)



Задача: внести следующие изменения в реализацию примера с кольцевой топологией сети:

- топология сети должна соответствовать представленному образцу
- 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.

## Упражнение

```
1 set ns [new Simulator]
2 $ns rtproto DV
3 set nf [open out.nam w]
4 $ns namtrace-all $nf
5 set f [open out.tr w]
6 $ns trace-all $f
7 proc finish {} {
8     global ns f nf
9     $ns flush-trace
10    close $f
11    close $nf
12    exec nam out.nam &
13    exit 0
14 }
15 set N 6
16 for {set i 0} {$i < $N} {incr i} {
17     set n($i) [$ns node]
18 }
19 for {set i 0} {$i < ($N-1)} {incr i} {
20     $ns duplex-link $n($i) $n([expr ($i+1)%($N-1)]) 1Mb 10ms DropTail
21 }
22 $ns duplex-link $n(5) $n(1) 1Mb 10ms DropTail
23
24 set tcp [new Agent/TCP/Newreno]
25 $ns attach-agent $n(0) $tcp
26 set ftp [new Application/FTP]
27 $ftp attach-agent $tcp
28 set sink [new Agent/TCPSink/DelAck]
29 $ns attach-agent $n(5) $sink
30 $ns connect $tcp $sink
31 $ns at 0.5 "$ftp start"
32 $ns rtmodel-at 1.0 down $n(0) $n(1)
33 $ns rtmodel-at 2.0 up $n(0) $n(1)
34 $ns at 4.5 "$ftp stop"
35 $ns at 5.0 "finish"
```

Результат: (рис. @fig:009)

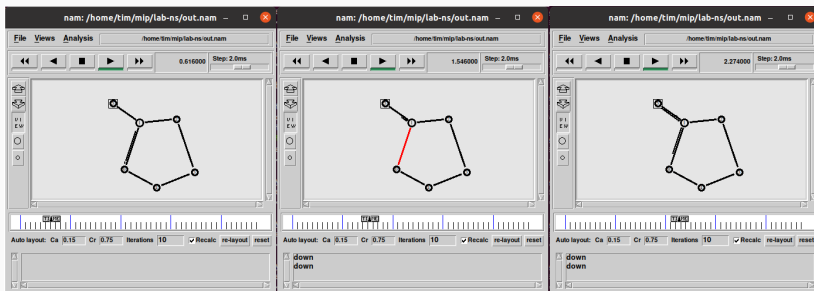


Рис. 9: модель2

## Выводы

---

- Приобрел навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2
- Проанализировал полученные результаты моделирования.