

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:

Ибрагимов Улугбек Ботырхонович

Группа:

НФИбд-02-20

МОСКВА

2023 г.

Цель	3
Задачи	4
Выполнение лабораторной работы.....	5
Создание шаблона сценария для NS-2	5
Выполнение примера №1	5
Выполнение примера №2.....	7
Выполнение примера №3.....	10
Выполнение упражнения	12
Анализ результатов	14
Вывод.....	15

Цель

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

Задачи

1. Создать шаблон сценария для NS-2.
2. Смоделировать сети передачи данных из примеров, данных в материалах к лабораторной работе.
3. Внести следующие изменения в реализацию примера с кольцевой топологией сети:
 - топология сети должна соответствовать представленной на рис. 1;
 - 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.

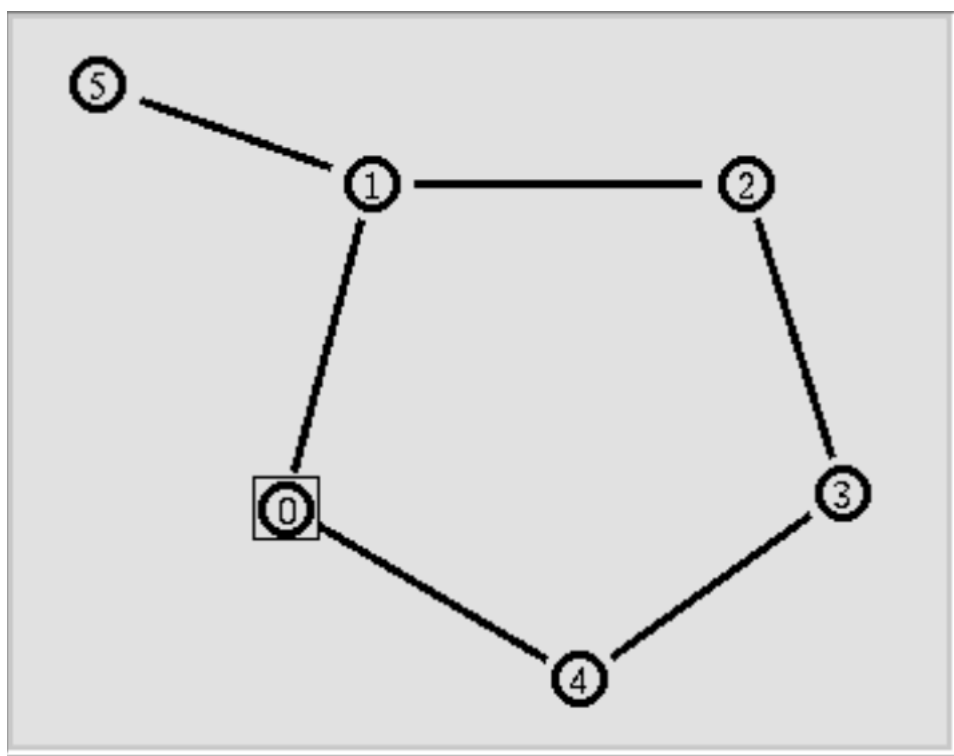


Рис.1 Изменённая кольцевая топология сети

Выполнение лабораторной работы

Создание шаблона сценария для NS-2

1. Создадим шаблон сценария для NS-2. В данном шаблоне создадим объект типа Simulator.
2. Также создадим переменные для записи в nam-файл выходных результатов моделирования, для записи в файл трассировки всех событий модели.
3. После этого добавим процедуру *finish*, которая закрывает файлы трассировки и запускает nam.
4. С помощью команды *at* указываем планировщику событий, что процедуру *finish* следует запустить через 5 с после начала моделирования, после чего запустить симулятор *ns*.

```
set ns [new Simulator]
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
}

$ns at 5.0 "finish"
$ns run
```

Выполнение примера №1

Описание сети:

Сеть передачи данных, состоит из двух узлов, соединённых дуплексной линией связи с полосой пропускания 2 Мб/с и задержкой 10мс, 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. Возьмем за основу, написанный шаблон. Перед строкой [`$ns at 5.0 «finish»`] будем описывать сеть. Сначала опишем топологию сети:

```
set N 2
for {set i 0} {$i < $N} {incr i} {
    set n($i) [$ns node]
}
$ns duplex-link $n(0) $n(1) 2Mb 10ms DropTail
```

2. Создадим агента для генерации трафика. Для этого создадим агента UDP и прикрепим его к нулевому узлу. После этого создадим и опишем источник трафика и прикрепим его к агенту UDP:

```
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
```

3. Создадим также Null-агента, который работает как приёмник трафика, и прикрепим его к узлу n1:

```
set null0 [new Agent/Null]
$ns attach-agent $n(1) $null0
```

4. Соединим агентов между собой. Добавим at-события в планировщик событий для запуска и остановки приложения CBR.

```
$ns connect $udp0 $null0

$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop»
```

5. Запустим, получившийся скрипт в консоли при помощи команды "`ns filename.tcl`". Откроется аниматор nam, где можно просмотреть симуляцию (Рис. 2).

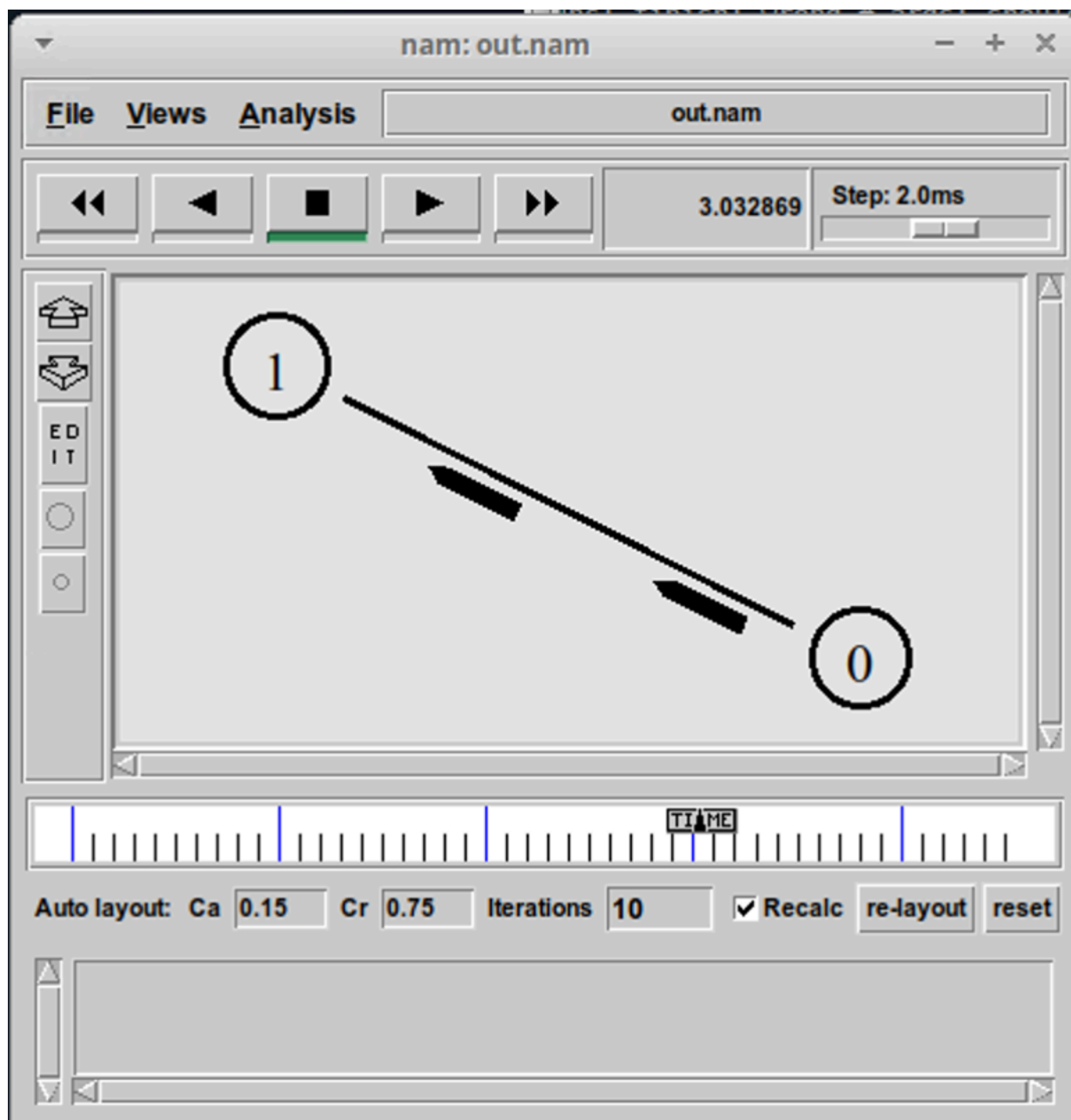


Рис. 2. Визуализация сети из примера №1.

Выполнение примера №2

Описание 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 10мс;
- 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. Возьмем за основу, написанный шаблон. Перед строкой [`$ns at 5.0 «finish»`] будем описывать сеть. Сначала опишем топологию сети:

```
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(3) $n(2) 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
```

2. Создадим агент UDP с прикреплённым к нему источником CBR и агент TCP с прикреплённым к нему приложением FTP:

```
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
```



```
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
set ftp [new Application/FTP]
$ftp attach-agent $tcp1
```

3. Создадим агенты-получатели:

```
set null0 [new Agent/Null]
$ns attach-agent $n(3) $null0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n(3) $sink1
```

4. Соединим агенты udp0 и tcp1 и их получателей. Зададим описание цвета каждого потока. Также добавим отслеживание событий в очереди и наложим ограничения на размер очереди:

```
$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
```

5. Добавим at-события и запустим симуляцию (Рис. 3):

```
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr0 stop"
```

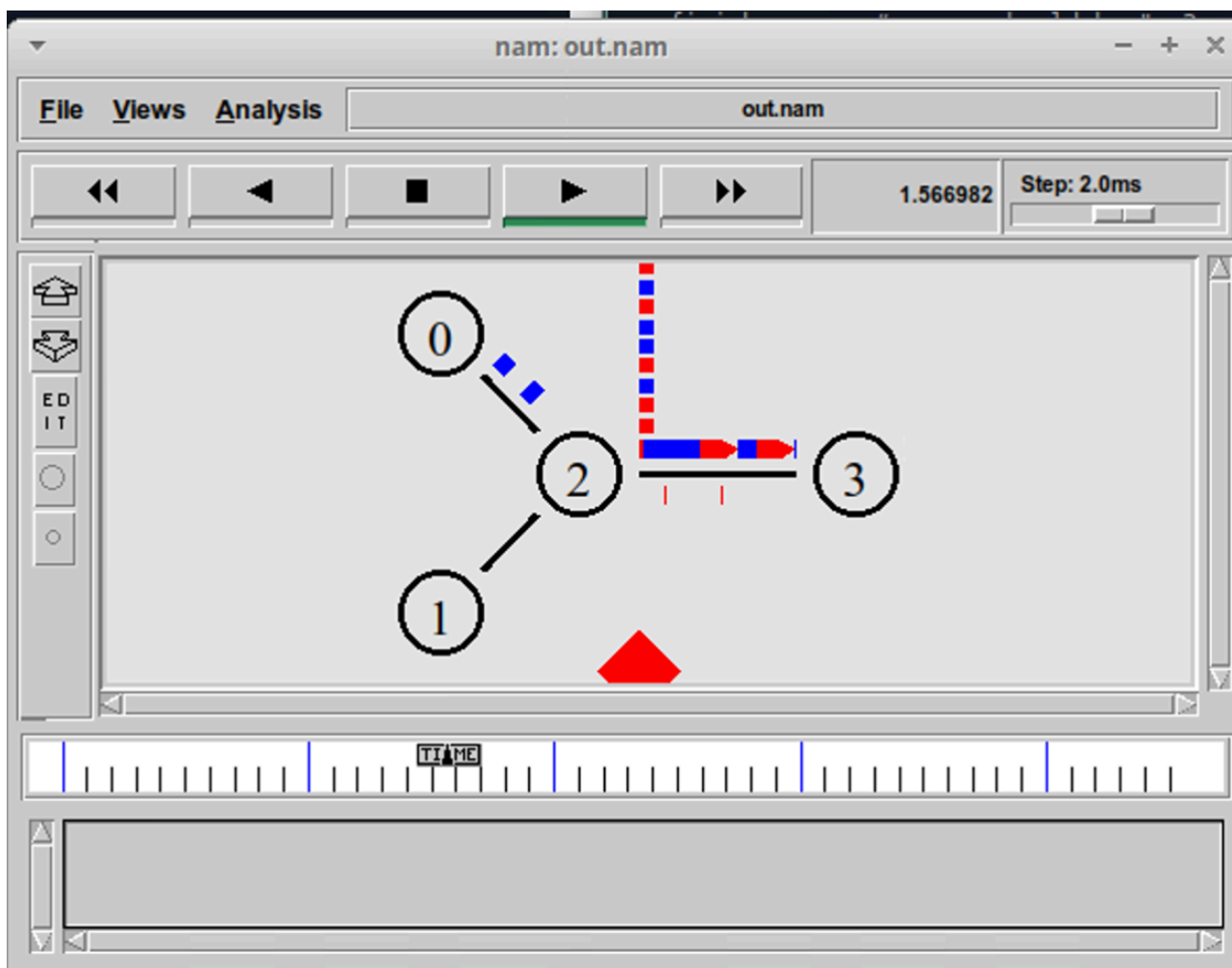


Рис. 3. Визуализация сети из примера №2.

Выполнение примера №3

Описание сети:

- 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.

1. Возьмем за основу, написанный шаблон. Перед строкой [`$ns at 5.0 «finish»`] будем описывать сеть. Сначала опишем топологию сети:

```
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
}

```

2. Зададим передачу данных от узла $n(0)$ к узлу $n(3)$:

```

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

```

3. Добавим at-события и команду разрыва соединения между узлами $n(1)$ и $n(2)$ на время в одну секунду:

```

$ns at 0.5 "$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»

```

4. Запустим симуляцию (Рис. 4 - 5):

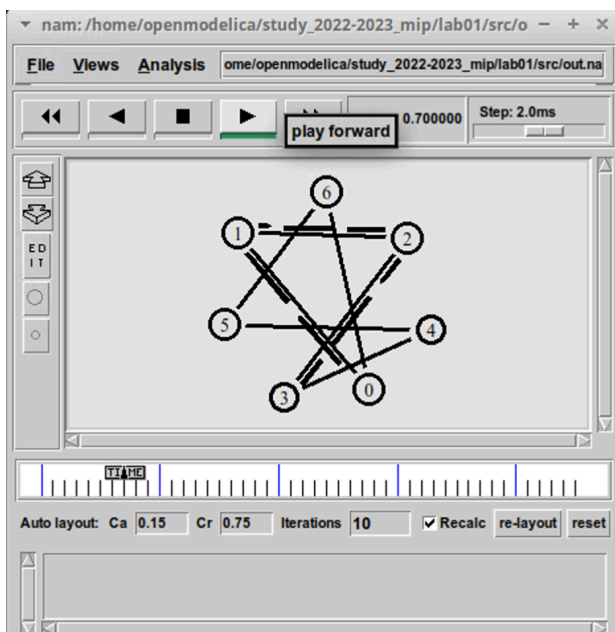


Рис. 4. Визуализация сети из примера №3.

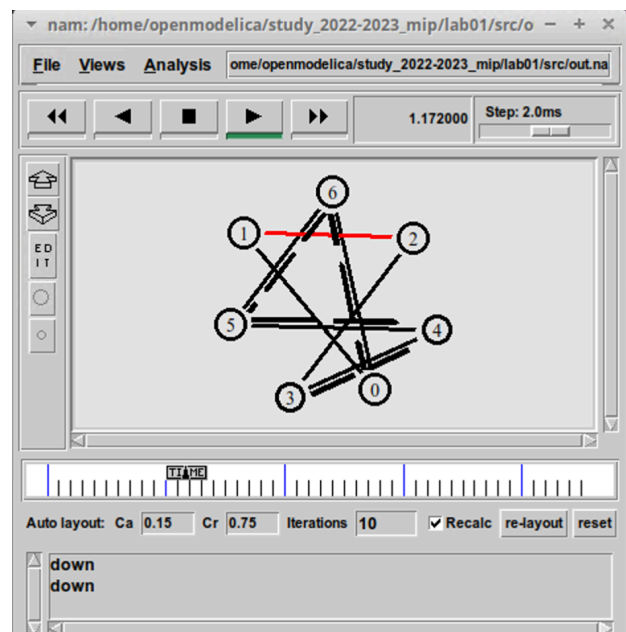


Рис. 5. Визуализация сети из примера №3. (Разрыв)

Выполнение упражнения

1. Возьмем за основу, написанный шаблон. Перед строкой [`$ns at 5.0 «finish»`] будем описывать сеть. Сначала опишем топологию сети. Создадим кольцевое подключение между узлами с `n(0)` по `n(4)`, после чего создадим соединение между узлами `n(1)` и `n(5)`:

```
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 ($i+1)%($N - 1)]) 1Mb
    10ms DropTail
}
$ns duplex-link n(1) n(5) 1Mb 10ms DropTail
```

2. Создадим агента `tcp` и прикрепим его к узлу `n(0)`. Также к агенту `tcp` прикрепим `FTP` и источник графика `CBR`. В качестве принимающей стороны создадим `TCPSink` и прикрепим его к узлу `n(5)`. После чего создадим подключение:

```
set tcp0 [new Agent/TCP]
$ns attach-agent n(0) $tcp0
set ftp [new Application/FTP]
$ftp attach-agent $tcp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $tcp0

set sink0 [new Agent/TCPSink]
$ns attach-agent n(5) $sink0

$ns connect $tcp0 $sink0
```

3. Добавим `at`-события и разрыв соединения между узлами `n(0)` и `n(1)`:

```
$ns at 0.5 "$cbr0 start"
$ns rtmodel-at 1.0 down n(0) n(1)
$ns rtmodel-at 2.0 up n(0) n(1)
$ns at 4.5 "$cbr0 stop"
```

4. Чтобы когда соединение будет разорвано, информация о топологии обновилась, и пакеты отсылались по новому маршруту. Добавим в начало файла (после объявления симулятора):

```
$ns rtproto DV
```

5. Запустим симуляцию (Рис. 6-7):

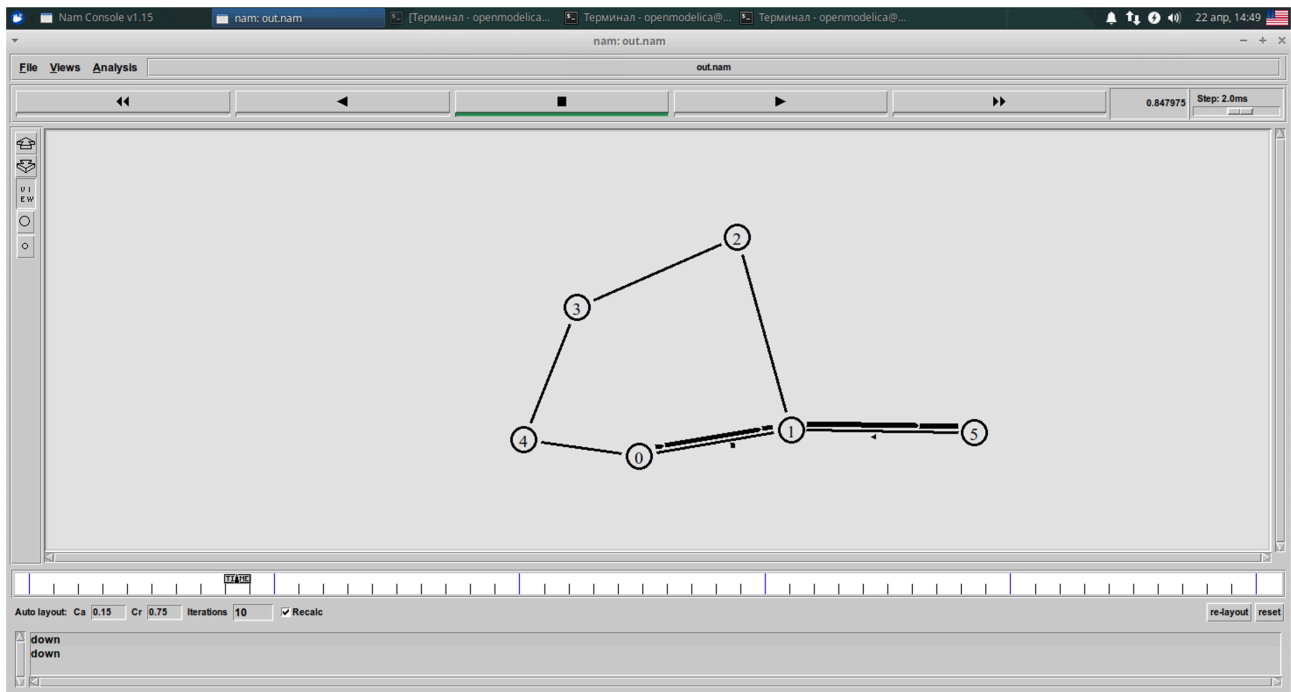


Рис. 6. Визуализация сети из упражнения.

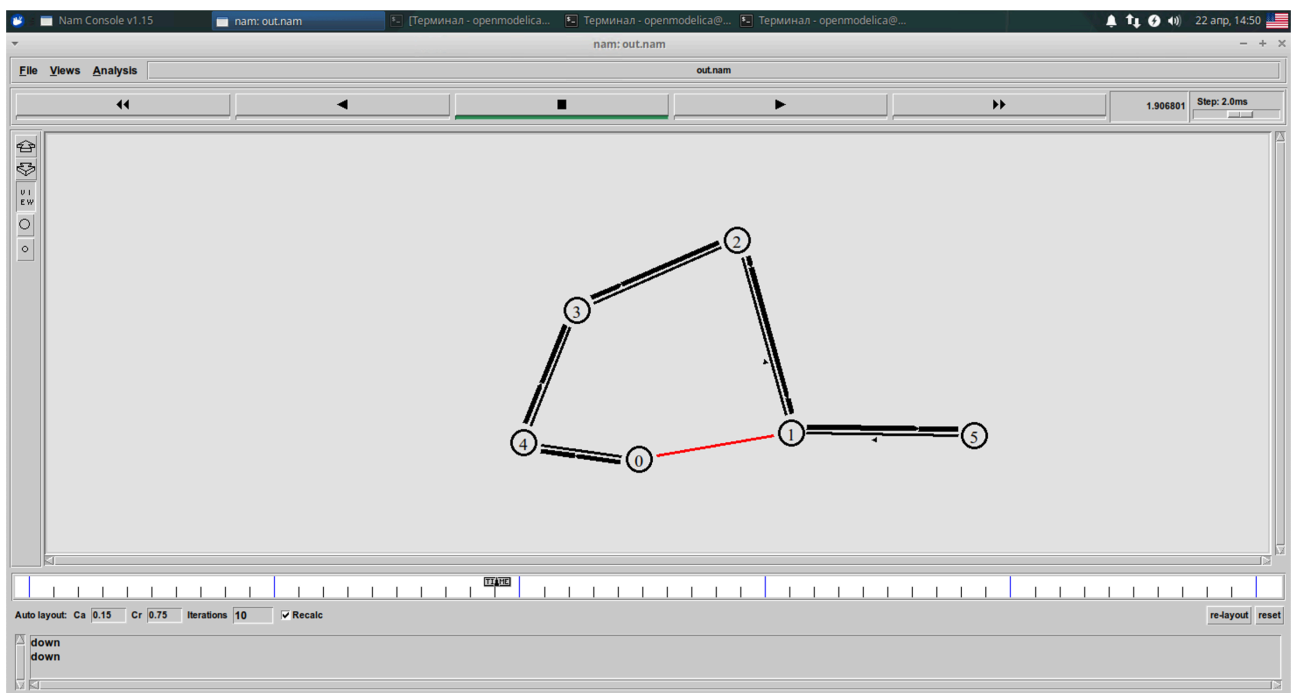


Рис. 7. Визуализация сети из упражнения (разрыв).

Анализ результатов

В рамках выполнения работы произошли некоторые незначительные проблемы, но все они решены. В остальном работа выполнена без непредвиденных проблем в соответствии с руководством: ошибок и сбоев не произошло.

Стоит отметить, что написание скриптов NS-2 происходи крайне комфортно, синтаксис, объекты интуитивно понятны. Также сами скрипты довольно быстро выполняются.

Аниматор `nim`, также оказался удобен для пользования, он отлично отображает топологию и позволяют смотреть симуляцию работы сети. Также позволяет вводит некоторые корректировки для удобства просмотра.

Вывод

В результате выполнения работы, были получены и улучшены практические навыки моделирования на NS-2. Также были созданы, настроены и смоделированы несколько сетей от простых до более сложных.