

**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: Николаев Александр Викторович

Группа: НФИбд-01-17

**МОСКВА**

2020 г.

## Цель работы

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

## Пример 1

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.

## Результат

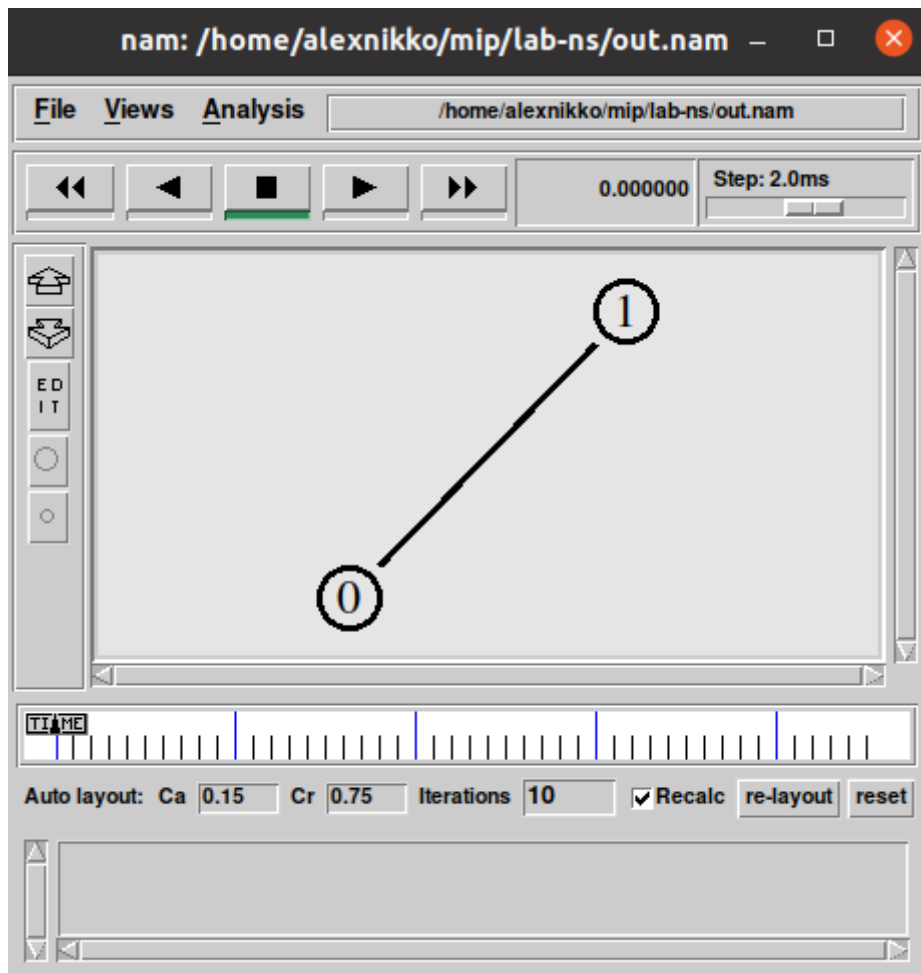


Рисунок 1. Пример простейшей модели сети

## Листинг

```
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
}

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

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 null0 [new Agent/Null]
$ns attach-agent $n(1) $null0
$ns connect $udp0 $null0

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

$ns at 5.0 "finish"
$ns run

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

## Пример 2

Постановка задачи: описание моделируемой 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 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;
- ТСП-источник на узле n0 подключается к ТСП-приёмнику на узле n3 (по умолчанию, максимальный размер пакета, который ТСП-агент может генерировать, равняется 1KByte);
- ТСП-приёмник генерирует и отправляет АСК 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 ТСП 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.

## Результат

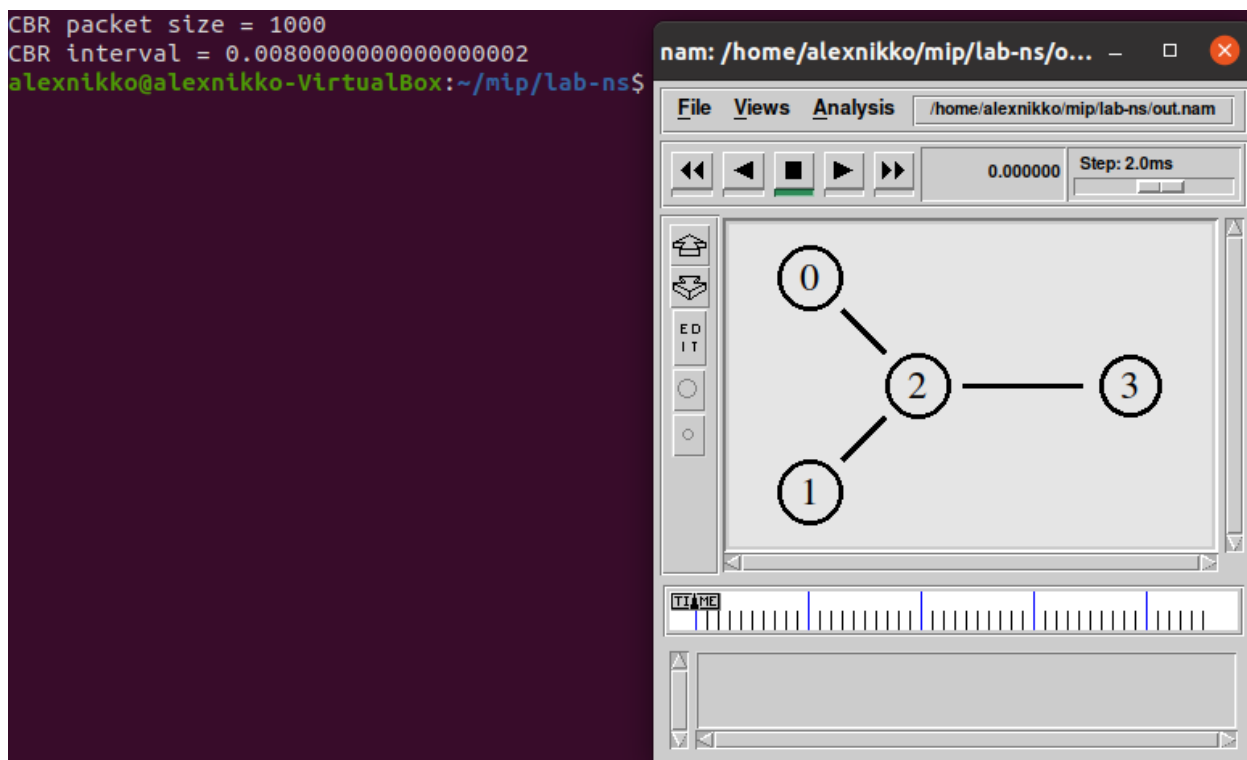


Рисунок 2. Пример усложненной модели сети.

## Листинг

```

#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
  
```

```

$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Open the all-trace file
set f [open out.tr w]
$ns trace-all $f

#Define a 'finish' procedure
proc finish {} {
    global ns nf f
    $ns flush-trace
    close $nf
    close $f
    exec nam out.nam &
    exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

```

```

#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

#Schedule events for CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation
$ns run

```

### Пример 3. Топология кольцо

Постановка задачи. 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.

## Результат

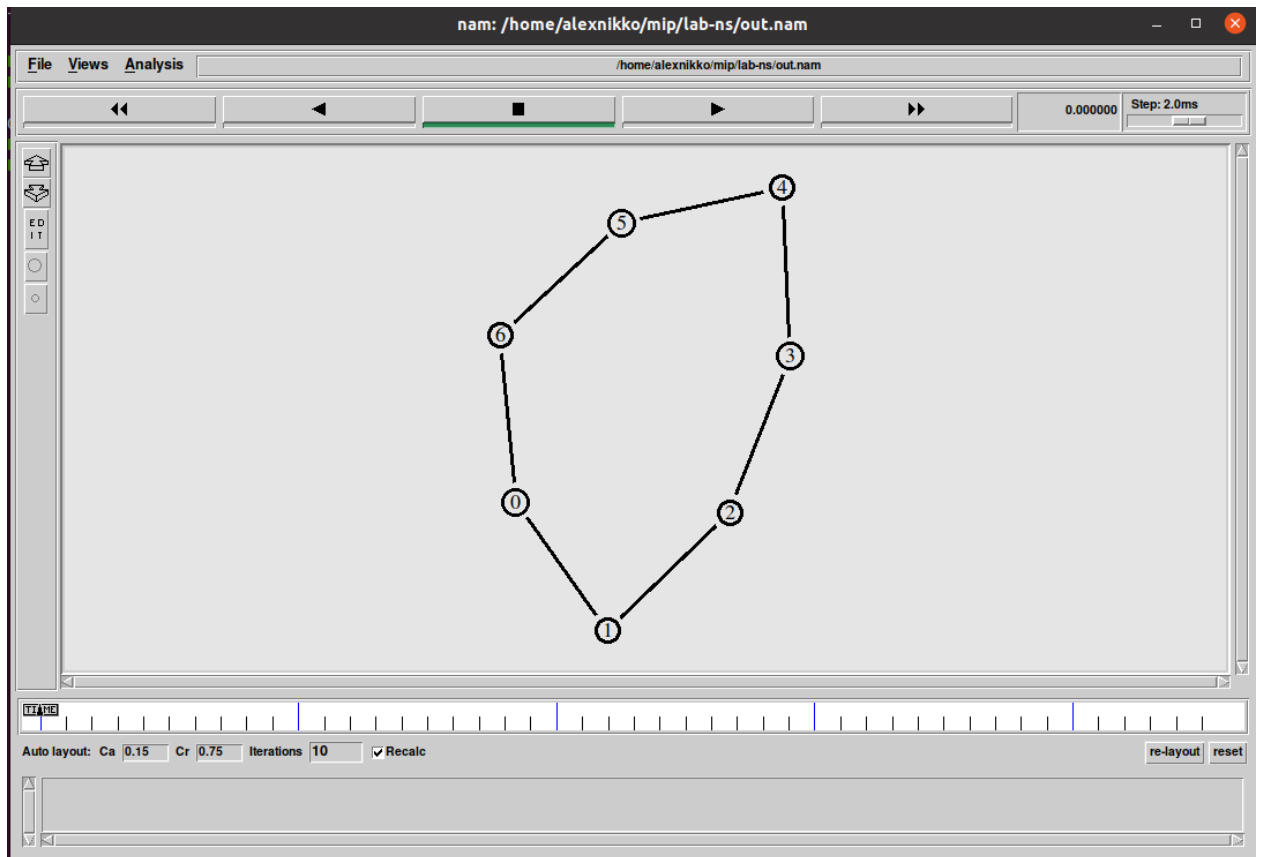


Рисунок 3. Пример топологии кольцо

## Листинг

```
set ns [new Simulator]

$ns rtproto DV

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
}

set N 7
for {set i 0} {$i < $N} {incr i} {
    set n($i) [$ns node]
}

for {set i 0} {$i < $N} {incr i} {
    set to [expr ($i+1)%$N]
}
```

```

        $ns duplex-link $n($i) $n($to) 1Mb 10ms DropTail
    }

    set udp [new Agent/UDP]
    $ns attach-agent $n(0) $udp
    set cbr [new Application/Traffic/CBR]
    $cbr attach-agent $udp
    $cbr set packet_size_ 500
    $cbr set interval_ 0.005

    set null [new Agent/Null]
    $ns attach-agent $n(3) $null
    $ns connect $udp $null

    $ns at 0.5 "$cbr start"
    $ns rtmodel-at 1.0 down $n(1) $n(2)
    $ns rtmodel-at 2.0 up $n(1) $n(2)
    $ns at 4.5 "$cbr stop"

    $ns at 5.0 "finish"
    $ns run

```

## Упражнение

Постановка задачи. Изменить предыдущую модель следующим образом:

- топология сети должна соответствовать представленной в задании к лабораторной работе;
- 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.



## Результат

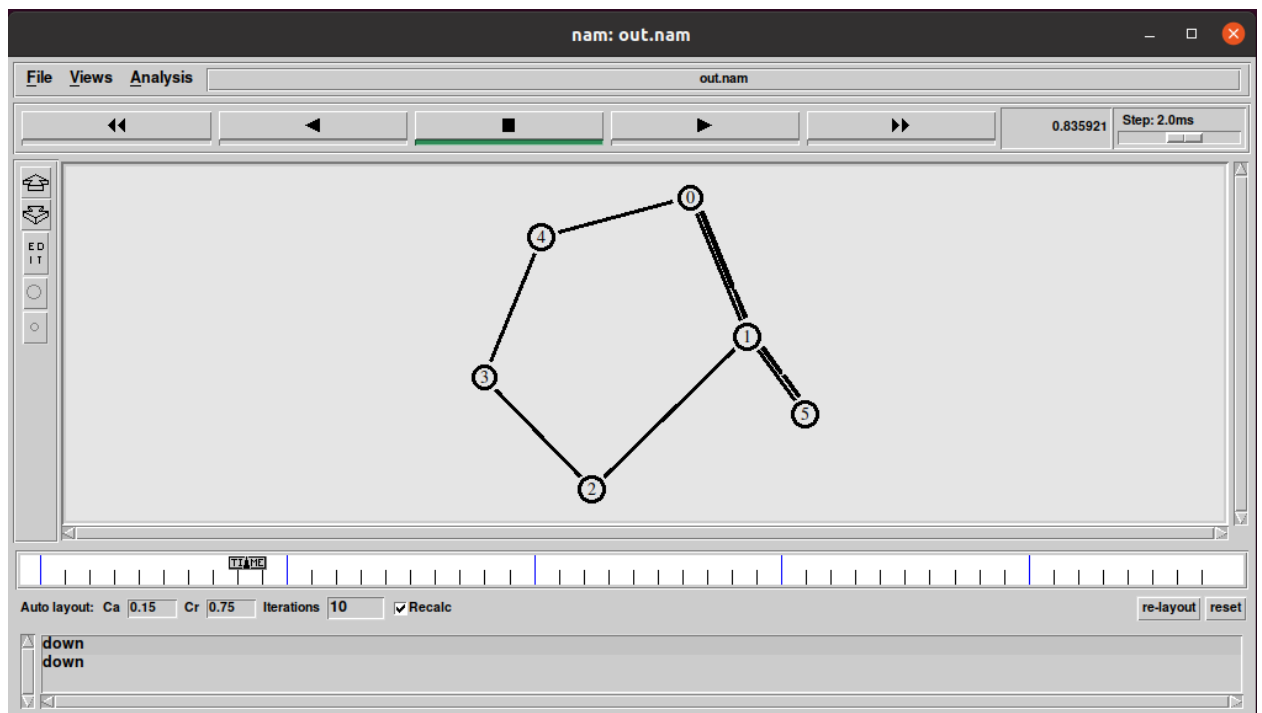


Рисунок 4. Начало передачи пакетов в полученной сети.

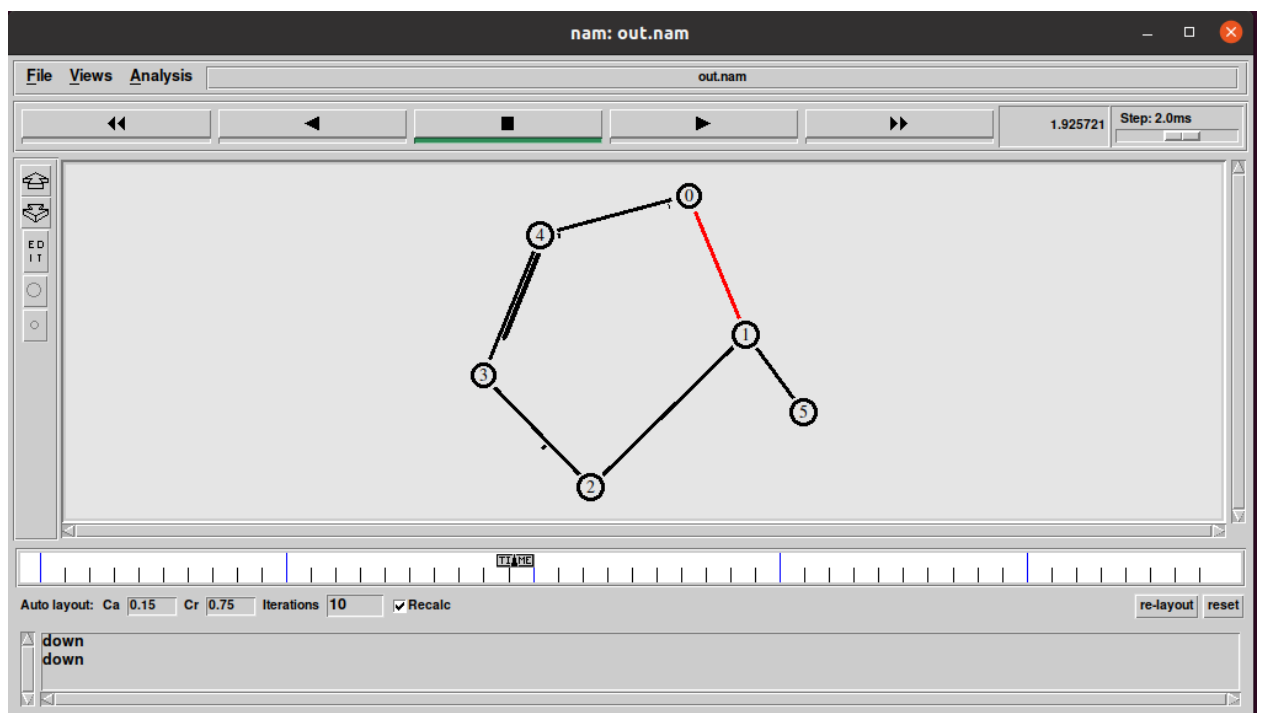


Рисунок 5. Передача пакетов после разрыва соединения.

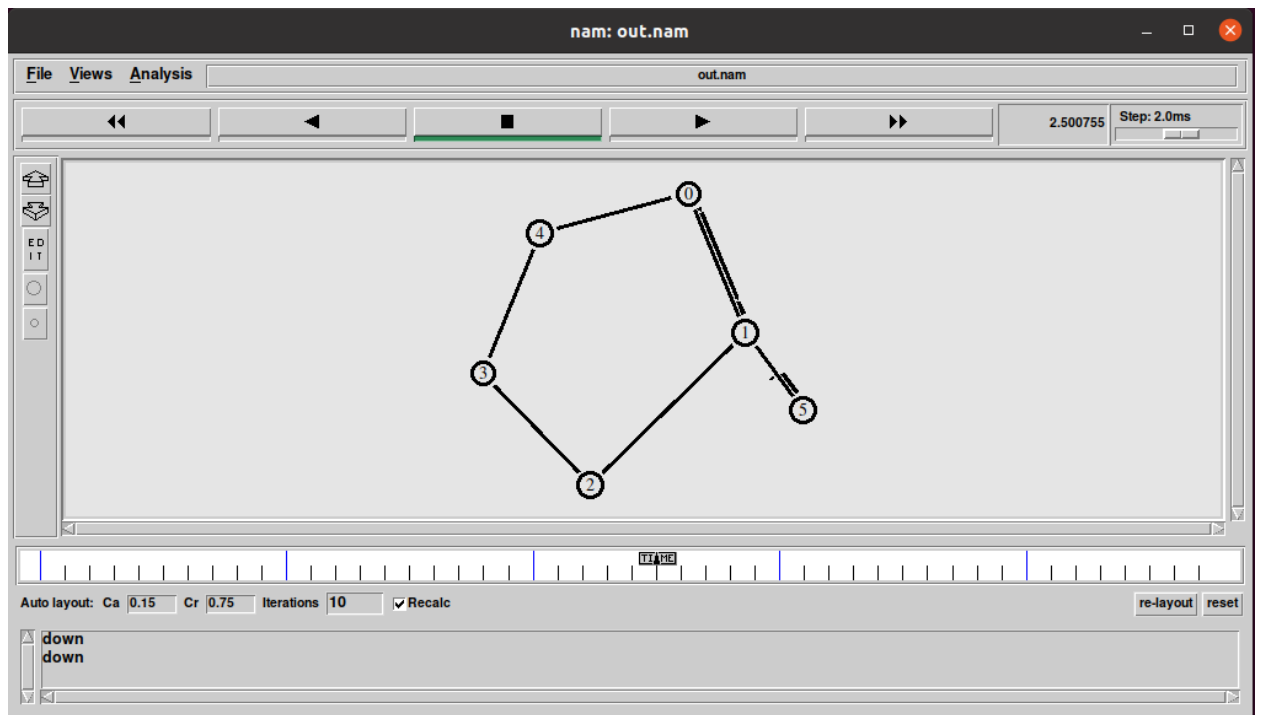


Рисунок 6. Передача пакетов после восстановления соединения.

## Листинг

```
set ns [new Simulator]

$ns rtproto DV

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
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n3 $n4 1Mb 10ms DropTail
$ns duplex-link $n0 $n4 1Mb 10ms DropTail
$ns duplex-link $n1 $n5 1Mb 10ms DropTail
```

```

set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n5 $sink
$ns connect $tcp $sink

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

$ns at 0.5 "$ftp start"
$ns rtmodel-at 1.0 down $n0 $n1
$ns rtmodel-at 2.0 up $n0 $n1
$ns at 4.5 "$ftp stop"

$ns at 5.0 "finish"
$ns run

```

## Вывод

В ходе выполнения лабораторной работы мы научились пользоваться основными инструментами NS-2, строить простые модели топологий сети, в том числе и кольцевую. Посмотрели на поведение разных протоколов. Убедились, что при разрыве соединения, топология «кольцо» продолжает эффективно пересылать пакеты.

## Приложение. Листинг шаблона.

```

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

```