

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: Шалыгин Георгий

Эдуардович

Группа: НФИбд-02-20

МОСКВА

2023 г.

Постановка задачи

Цель работы: приобретение навыков моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2, а также анализ полученных результатов.

Задачи:

1. Создать шаблон сценария.
2. 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.
3. Смоделировать следующую сеть : – 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;
 - 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.
4. 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

5. Выполнить упражнение.

Выполнение работы

Создание шаблона

1. Создадим и откроем файл shablon.tcl.

```
openmodelica@openmodelica-VirtualBox:~$ mkdir -p mip/lab-ns
openmodelica@openmodelica-VirtualBox:~$ cd mip/lab-ns
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ touch shablon.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ emacs shablon.tcl
```

2. Опишем скрипт для шаблона моделирования

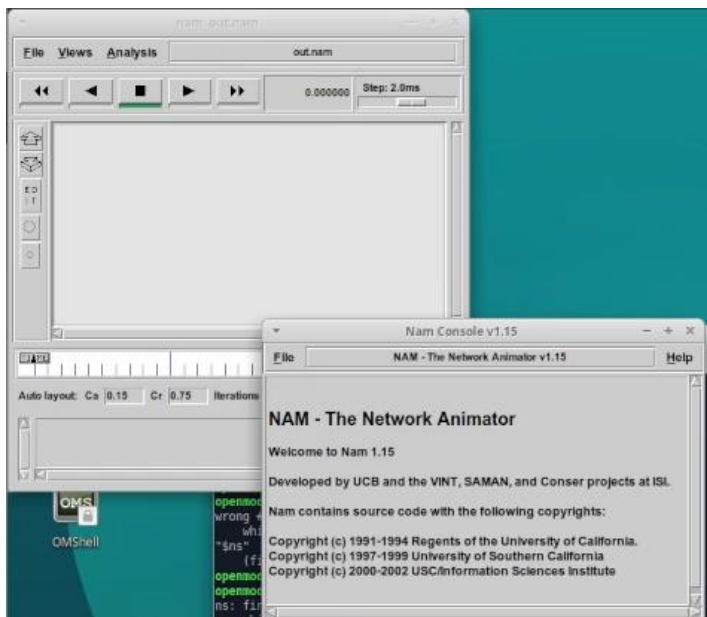
```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f
proc finish {} {
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam out.nam &
    exit 0
}
$ns at 5.0 "finish"
$ns run
```

Здесь:

- создается объект симуляции Simulator
- создается переменная nf для записи результатов в nam-файл.
- переменная f для регистрации всех событий в trace-файл

- описывается процедура `finish`, выполняющаяся в конце симуляции, закрывающая файлы, прекращающая трассировку и запускающая `nam`.
- командой `at` описываются выполняемые события (в аднном случае только вызов `finish` на 5 секунде).
- запуск модели командой `run`.

3. После запуска видим пустую симуляцию



Создание модели из 2х узлов

4. Создаем файл для первой симуляции:

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp shablon.tcl example1.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ emacs example1.tcl
```

5. В файле описываем скрипт для моделирования сети из двух узлов

```

set n0 [$ns node]
set n1 [$ns node]

$ns duplex-link $n0 $n1 2Mb 10ms DropTail

set udp0 [new Agent/UDP]
$ns attach-agent $n0 $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 $n1 $null0

$ns connect $udp0 $null0

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

```

В этом файле:

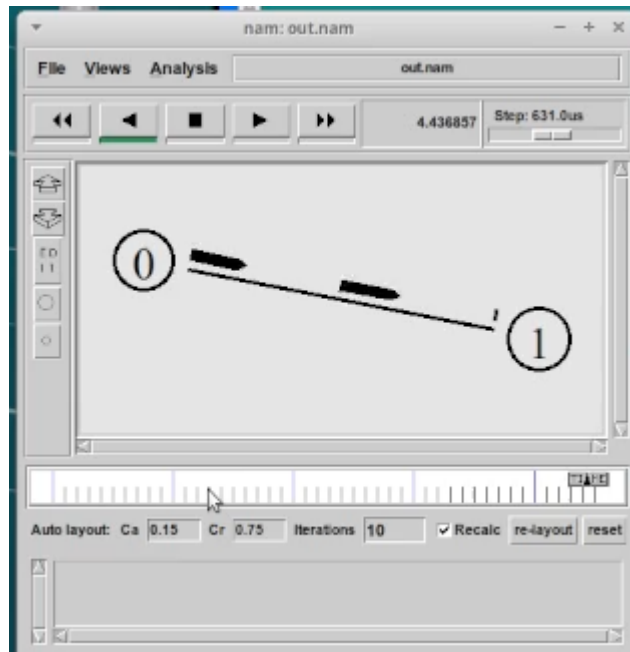
- создаются два узла n0, n1 с duplex связью на 2Mb/s, 10мс задержки и типов обслуживания DropTail
- создаются агенты udr на узле n0 с источником трафика cbr (пакеты по 500 байт с интервалом 0.005с) и null для приема (и сброса) пакетов на узле n1
- агенты udr и null соединяются
- описываются события для запуска передачи пакетов на 0.5с и остановки на 4.5с

6. После запуска симуляции можем посмотреть визуализацию:

```

openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ ns example1.tcl

```



Здесь отображена переча пакетов по UDP из 0 к 1 узлу.

Создание модели сети из 3х узлов

7. Создадим файл example2

```
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ cp example1.tcl example2.tcl
openmodelica@openmodelica-VirtualBox:~/mip/lab-ns$ emacs example2.tcl
```

8. Скрипт для второй модели:

- создадим 3 узла и опишем соединения

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n3 $n2 2Mb 10ms DropTail
```

- повесим агент udr на n0, подключим к нему cbr генератор трафика (пакет по 500 через каждый 0.005с), к n1 подключим tcp с ftp. Для приема создадим агенты null и sink. Сконнектим null с udr, sick с tcp.

```

set udp0 [new Agent/UDP]
$ns attach-agent $n0 $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 $n1 $tcp1

set ftp [new Application/FTP]
$ftp attach-agent $tcp1

set null0 [new Agent/Null]
$ns attach-agent $n3 $null0

set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1

$ns connect $udp0 $null0
$ns connect $tcp1 $sink1

```

- Зададим цвета потоков (красный tcp, синий udp), повесим очередь на связь n2-n3 с лимитом 20 пакетов. Опишем порядок действий симуляции: запуск udp с 0.5с до 4.5с, ftp работает с 1с до 4с.

```

$ns color 1 Blue
$ns color 2 Red

$udp0 set class_ 1
$tcp1 set class_ 2

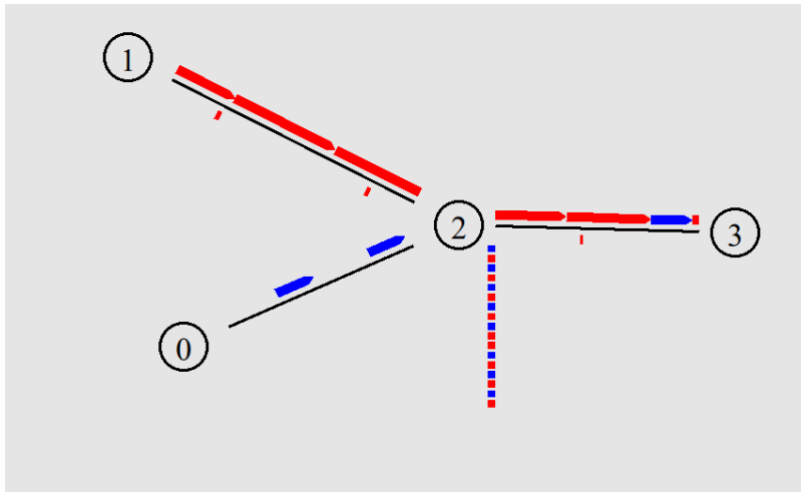
$ns duplex-link-op $n2 $n3 queuePos 0.5

$ns queue-limit $n2 $n3 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"

```

9. Запустим симуляцию описанной модели:



Видим красные пакеты tcp (с откликами) и синие udr. Также постепенно заполняемую очередь, из которой пакеты не входящие в лимит будут сброшены.

Сеть с кольцевой топологией.

10. Опишем модель сети в файле example3:

- Создадим 7 узлов с описанием связей:

```
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
}
```

- Повесим на 0 узел udr с cbr, на 3 агент приемник null, и свяжем их.

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

set cbr0 [new Agent/CBR]
$ns attach-agent $n(0) $cbr
$cbr0 set packetSize 500
$cbr0 set interval_ 0.005

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

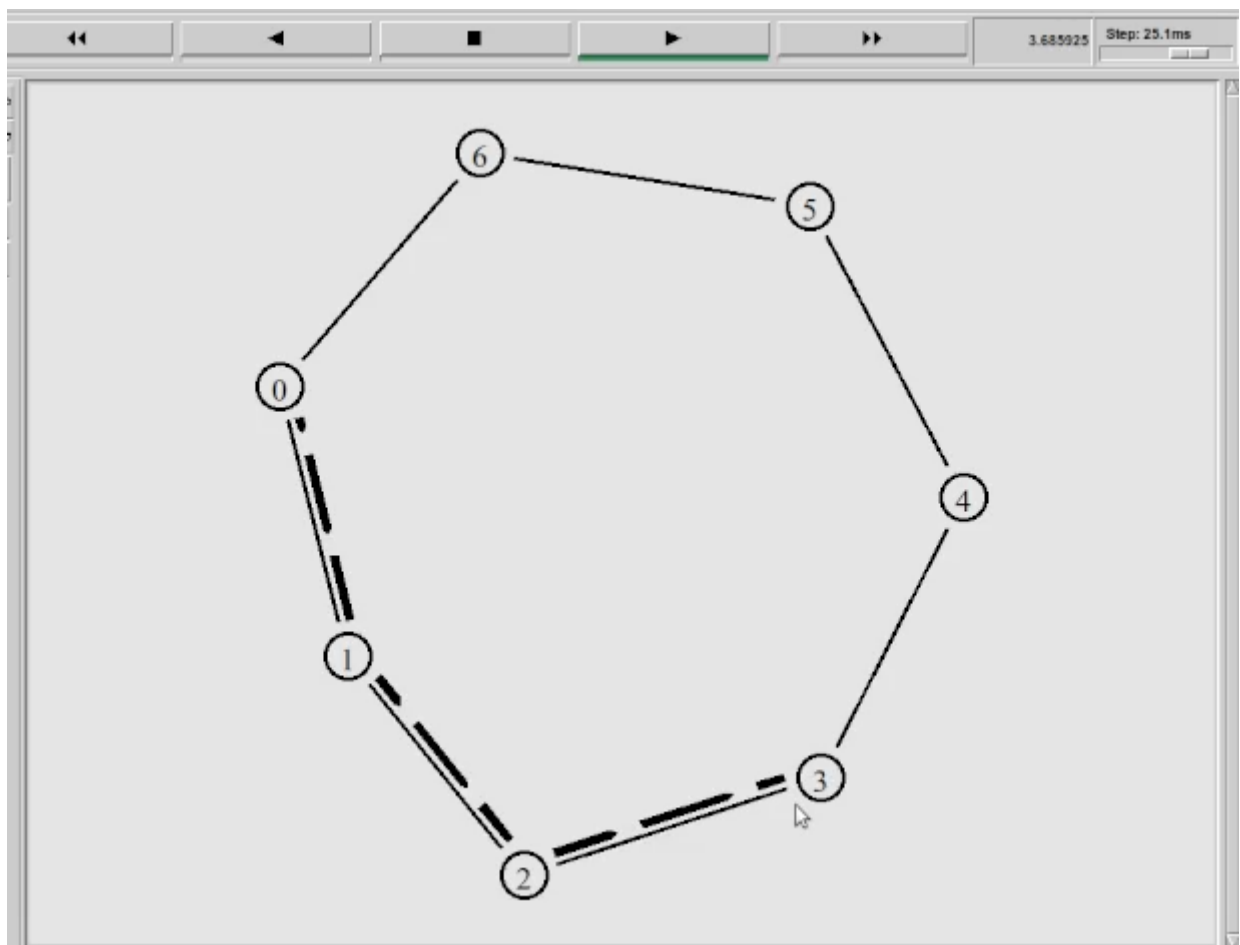
$ns connect $cbr0 $null0
```

- Опишем порядок запуска:

```
$ns at 0.5 "$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"
$ns run
```

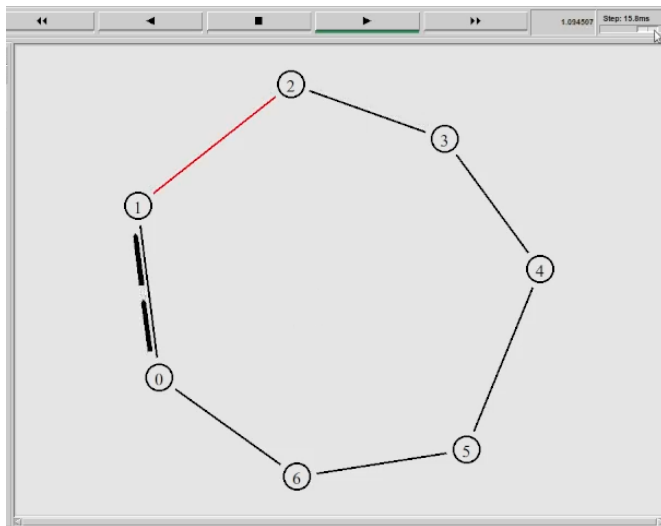

11. В результате пакеты идут по кратчайшему маршруту:



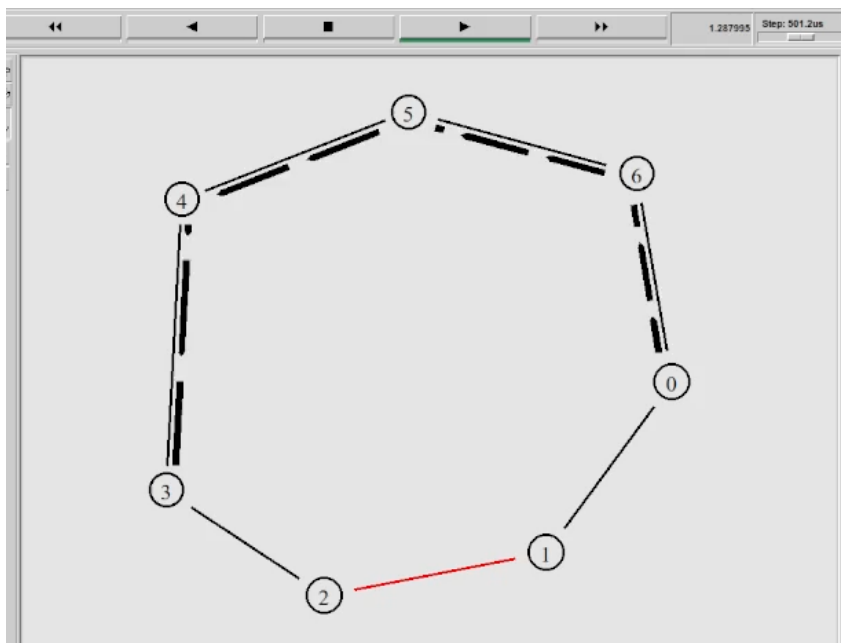
12. Добавим разрыв связи 1-2

```
$ns at 0.5 "$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"
```

13. В результате после разрыва пакеты просто сбрасываются:



14. Добавим команду `$ns rproto DV` и запустим симуляцию:



В результате послыки пакетов обмена данными для маршрутизации, маршрут перестраивается на новый.

Упражнение.

15. Описание моделируемой сети:

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.

16. Опишем скрипт для описанной сети:

```
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

set tcpl [new Agent/TCP]
$ns attach-agent $n(0) $tcpl

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

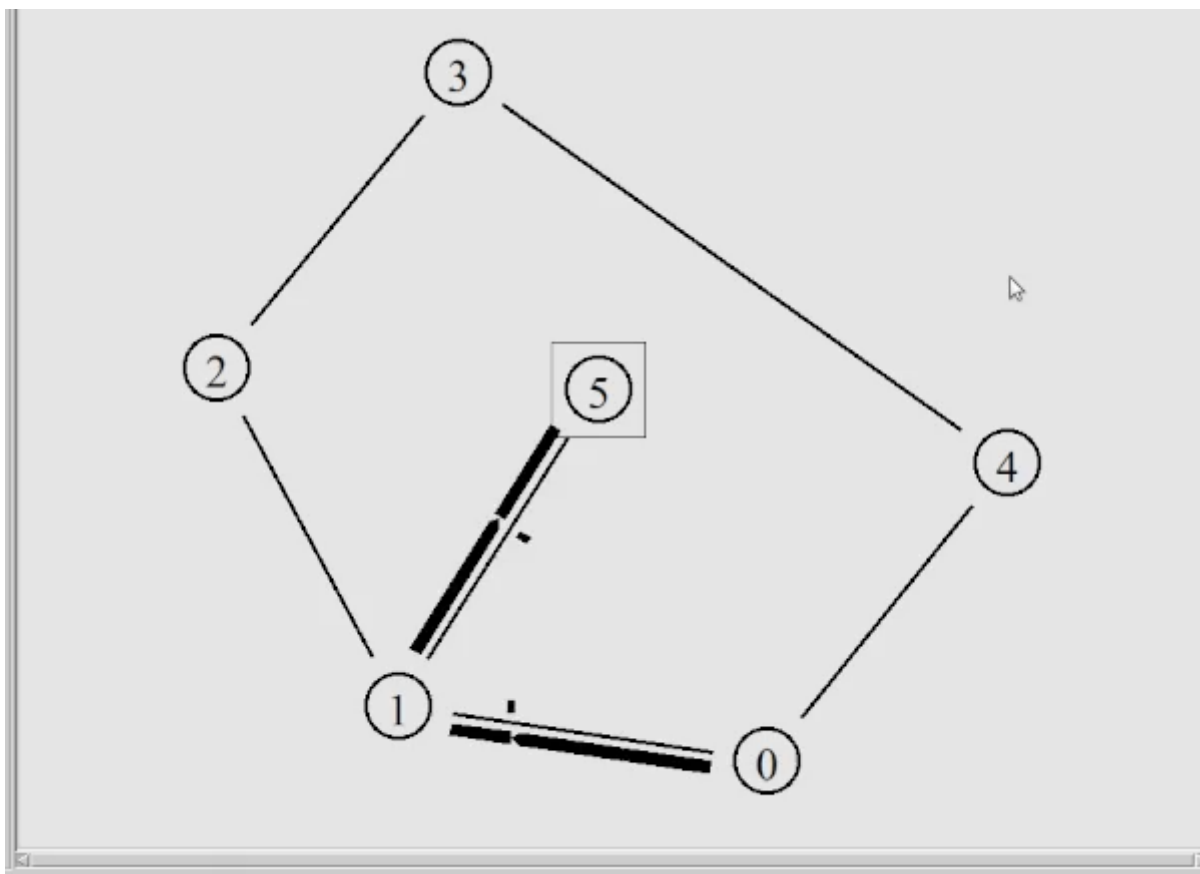
set sink1 [new Agent/TCPSink]
$ns attach-agent $n(5) $sink1

$ns connect $tcpl $sink1

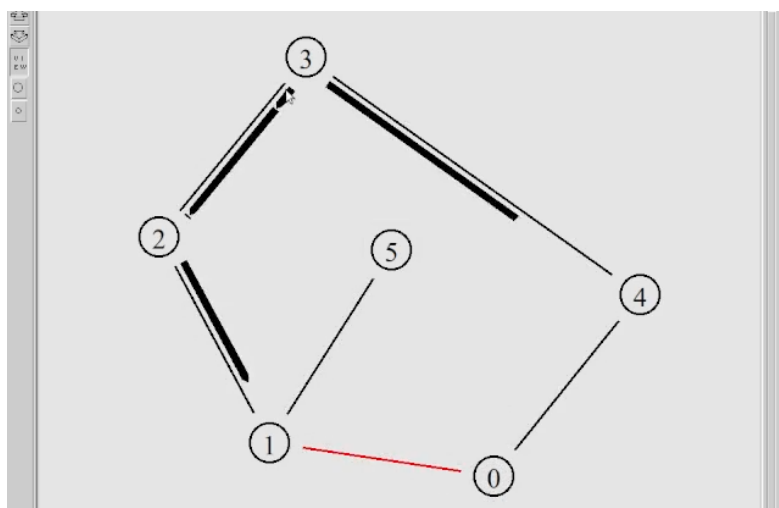
$ns at 0.5 "$ftp start"
$ns rtmodel-at 1.0 down $n(1) $n(0)
$ns rtmodel-at 2.0 up $n(1) $n(0)
$ns at 4.5 "$ftp stop"
```

17. Запустим симуляцию:

- сначала пакеты передаются по следующему маршруту:



- после разрыва маршрут перестраивается:



Заключение

В результате работы я приобрел навыки моделирования сетей передачи данных с помощью средства имитационного моделирования NS-2. Я создал шаблон сценария и описал с его помощью модели для сетей из 2-х узлов, 3-х узлов, топологии кольцо. Я научился описывать используемые для передачи пакетов протоколы, очереди, моделировать переполнение очереди и разрывы в сети и

анализировать поведение сети в результате этих событий.