Лабораторная работа №1

Простые модели компьютерной сети

Алиева Милена Арифовна

Российский университет дружбы народов, Москва, Россия

Содержание

Содержание

Цель

Цель

Целью данной работы является получение практических навыков работы в консоли с расширенными атрибутами файлов

Задание

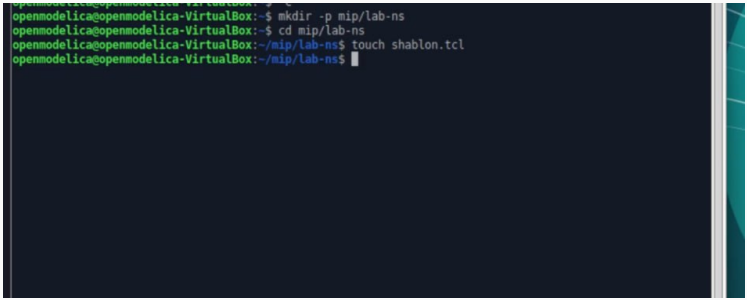
Задание

Познакомиться на примерах с тем, как используются основные и расширенные атрибуты при разграничении доступа

Порядок выполнения

Порядок выполнения

1. В своём рабочем каталоге создаём директорию mip, в которой будут выполняться лабораторные работы. Внутри mip создаём директорию lab-ns, а в ней файл shablon.tcl.



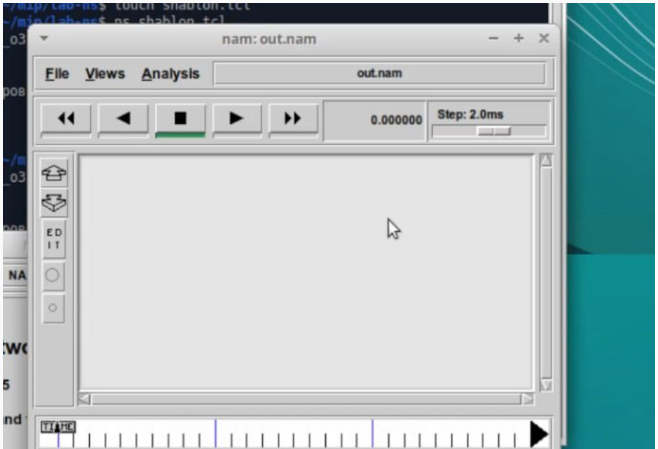
Порядок выполнения

2. Открываем на редактирование файл shablon.tcl. Сначала создадим объект типа Simulator, создадим переменную nf и укажем, что требуется открыть на запись nam-файл для регистрации выходных результатов моделирования, вторая строка даёт команду записывать все данные о динамике модели в файл out.nam. Далее создадим переменную f и откроем на запись файл трассировки для регистрации всех событий модели. После этого добавим процедуру finish, которая закрывает файлы трассировки и запускает nam. С помощью команды at указываем планировщику событий, что процедуру finish запустим через 5 с после начала моделирования, после чего запустим симулятор ns.

```
# создание объекта Simulator
set ns [new Simulator]

# открытие на запись файла out.nam для визуализатора nam
set nf [open out.nam w]
# все результаты моделирования будут записаны в переменную nf
$ns namtrace-all $nf

# открытие на запись файла трассировки out.tr
# для регистрации всех событий
set f [open out.tr w]
# все регистрируемые события будут записаны в переменную f
$ns trace-all $f

# процедура finish закрывает файлы трассировки
```

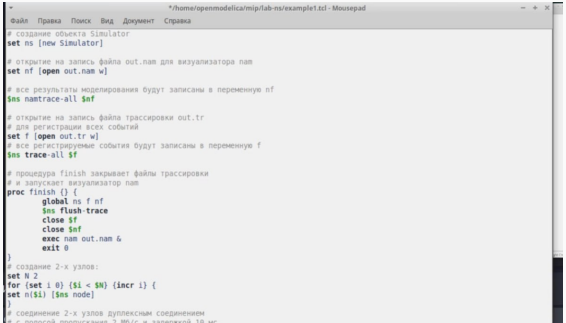
Порядок выполнения

3. Сохранив изменения в отредактированном файле shablon.tcl и закрыв его, запустим симулятор командой ns shablon.tcl. Увидим пустую область моделирования, поскольку ещё не определены никакие объекты и действия.



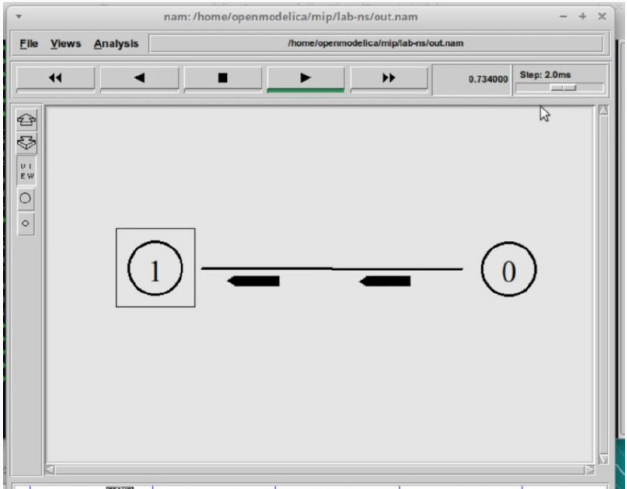
Порядок выполнения

4. Выполним второй пример, который посвящён описанию топологии сети, состоящей из двух узлов и одного соединения. Нам 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.



Порядок выполнения

5. Сохранив изменения в отредактированном файле и запустив симулятор, получим анимированный результат моделирования.



Порядок выполнения

6. Выполним третий пример. 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;
- TCP-источник на узле n0 подключается к TCP-приёмнику на узле n3 (по-умолчанию, максимальный размер пакета, который TCP-агент может генерировать, равняется 1KByte)

Порядок выполнения

Порядок выполнения

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.

Порядок выполнения

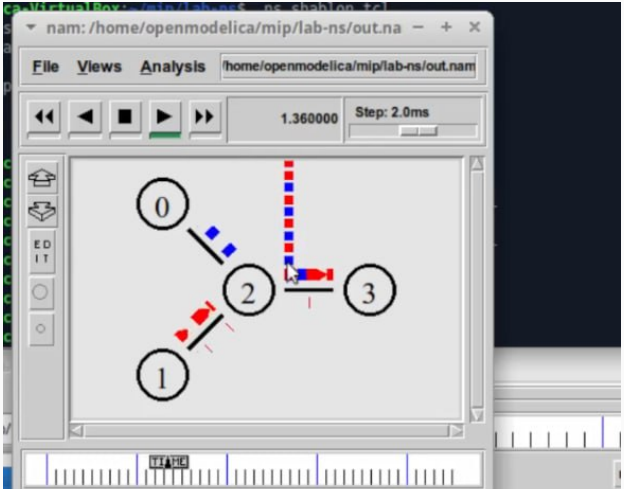
Порядок выполнения

Откроем example2.tcl на редактирование, создадим 4 узла и 3 дуплексных соединения с указанием направления, создадим агент UDP с прикреплённым к нему источником CBR и агент TCP с прикреплённым к нему приложением FTP, создадим агенты-получатели, соединим агенты udp0 и tcp1 и их получателей, зададим описание цвета каждого потока, выполним отслеживание событий в очереди и наложение ограничения на размер очереди, добавим at-события.

```
proc finish {} {
        global ns f nf
        $ns flush-trace
        close $f
        close $nf
        exec nam out.nam &
        exit 0
}
set N 4
for {set i 0} {$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) 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 n(0)
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 n(1)
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1

# создание приложения FTP
# и присоединение его к агенту tcp1
set ftp [new Application/FTP]
```

Порядок выполнения

7. Сохранив изменения в отредактированном файле и запустив симулятор, получим анимированный результат моделирования.



Порядок выполнения

Порядок выполнения

8. Описание модели 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.

Порядок выполнения

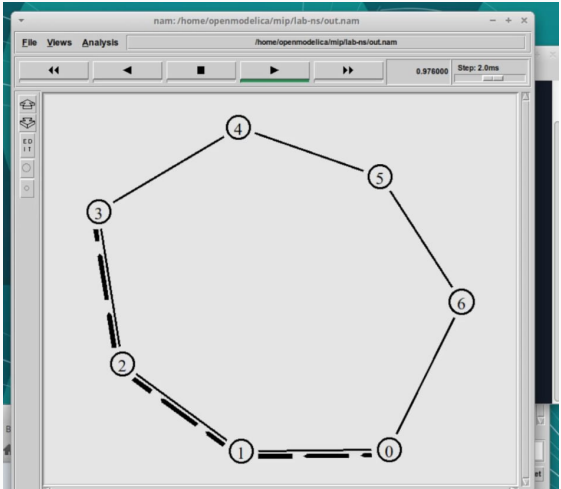
Порядок выполнения

Откроем example3.tcl на редактирование. Опишем топологию моделируемой сети, соединим узлы так, чтобы создать круговую топологию. Каждый узел, за исключением последнего, соединяется со следующим, последний соединяется с первым, для этого в цикле использован оператор %, означающий остаток от деления нацело. Зададим передачу данных от узла n(0) к узлу n(3). Данные передаются по кратчайшему маршруту от узла n(0) к узлу n(3), через узлы n(1) и n(2).

```
        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]) 1Mb 10ms DropTail
}

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)
```

Порядок выполнения

9. Сохранив изменения в отредактированном файле и запустив симулятор, получим анимированный результат моделирования



Порядок выполнения

Порядок выполнения

11. Выполним упражнение, для этого внесем следующие изменения в реализацию примера с кольцевой топологией 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.

Порядок выполнения

Порядок выполнения

```
set N 5
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
}

set n5 [$ns node]
$ns duplex-link $n5 $n(1) 1Mb 10ms DropTail

set tcp1 [new Agent/TCP/Newreno]
$ns attach-agent $n(0) $tcp1

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

set sink1 [new Agent/TCPSink/DelAck]
$ns attach-agent $n5 $sink1
$ns connect $tcp1 $sink1

$ns at 0.5 "$ftp start"
$ns rtmodel-at 1.0 down $n(0) $n(1)
$ns rtmodel-at 2.0 up $n(0) $n(1)
$ns at 4.5 "$ftp stop"
$ns at 5.0 "finish"
# at-событие для планировщика событий, которое запускает
```

Выводы

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.